

3100096008051

**ANALISA KAPASITAS KEKUATAN ULTIMATE  
DARI SAMBUNGAN BEAM DAN PELAT  
PADA TUBULAR MEMBER**

**TUGAS AKHIR**

RSke  
624.177.23  
Har  
a-1  
1996



PERPUSTAKAAN ITS	
Tgl. Terima	02 APR 1996
Terima Oleh	H
No. Agenda Dep.	6152.

Disusun oleh :

**WIDYO HARTONO**

**NRP. 4904300192**

**PROGRAM STUDI TEKNIK KELAUTAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1996**

**ANALISA KAPASITAS KEKUATAN ULTIMATE  
DARI SAMBUNGAN BEAM DAN PELAT  
PADA TUBULAR MEMBER**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Kelautan  
Pada  
Program Studi Teknik Kelautan  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
S u r a b a y a**

**Mengetahui / Menyetujui  
Dosen Pembimbing**

**(DR. Ir. D.M. ROSYID)**

**S U R A B A Y A  
1996**



FAKULTAS TEKNOLOGI KELAUTAN ITS  
PROGRAM STUDI TEKNIK LAUTAN

TUGAS - AKHIR.

No. : 019/PT12.H8.FTK4./Q/95

MATA KULIAH	:	TUGAS AKHIR .....
NOMOR MATA KULIAH	:	0E..1701.....
NAMA MAHASISWA	:	Widyo Hartono.....
NOMOR POKOK	:	490.4300.192.....
TANGGAL DIBERIKAN TUGAS	:	13. Februari. 1995.....
TANGGAL SELESAI TUGAS	:	13. Agustus. 1995.....
DOSEN PEMBIMBING	:	Dr., Ir., Daniel M. Rosyid.....

TEMA/URAIAN/DATA-DATA YANG DIBERIKAN :

" ANALISA KAPASITAS KEKUATAN ULTIMATE DARI  
SAMBUNGAN BEAM DAN PELAT PADA TUBULAR "

SURABAYA, ..13. Februari. 1995....

DIBUAT RANGKAP 4 :

1. Mahasiswa ybs.
2. Dekan
3. Dosen Pembimbing.
4. Arsip Kajur.



Dr. E. B. Djatmiko, MSc, PhD.

NIP. 131 407 592





# TUGAS AKHIR

ABSTRAK



## ABSTRAK

Beberapa penelitian telah menghasilkan penurunan persamaan kapasitas kekuatan *ultimate* dari sambungan beam dan pelat pada tubular member (IT dan PT *joint*). Pada penurunan persamaannya masih terdapat perbedaan antara yang satu dengan lainnya, hal ini masih juga merupakan suatu studi untuk mendapatkan persamaan baru yang lebih baik dan konsisten yang digunakan dalam evaluasi perancangan kekuatan sambungan struktur *truss* geladak bangunan lepas pantai. Permasalahan tersebut melatarbelakangi penulisan tugas akhir ini untuk menganalisa secara numerik dari kapasitas kekuatan *ultimate* sambungan IT dan PT dengan pendekatan teori statis non-linear pada lingkup konsep metode elemen hingga.

Analisa dilakukan dengan menggunakan suatu model elemen struktur dengan kondisi batas yang telah ditetapkan. Selanjutnya ditinjau pengaruh perubahan parameter geometri  $\beta$  pada sambungan terhadap kapasitas kekuatan *ultimate* sambungan akibat pembebanan aksial tekan. 5 jenis model sambungan IT dan 5 jenis model sambungan PT dengan nilai perubah parameter  $\beta$  masing-masing adalah 0.5, 0.6, 0.7, 0.8 dan 0.9 digunakan sebagai model analisa kapasitas kekuatan *ultimate* sambungan.

Sebagai verifikasi, hasil data pengujian atau  $(P_{ult})_{pengujian}$  dibandingkan dengan hasil analisa secara numerik atau  $(P_{ult})_{numerik}$ . Diperoleh hasil perbandingan adalah  $1.0 < \{(P_{ult})_{pengujian}\} + \{(P_{ult})_{numerik}\} < 1.05$ . Perbandingan ini menunjukkan bahwa hasil analisa  $P_{ult}$  dengan analisa statis non-linear lebih mendekati terhadap hasil pengujian dan hasil penurunan rumus Kurobane, dibandingkan dengan hasil penurunan Wardenier dan Kamba. Pengaruh perubahan parameter  $\beta$  terhadap kapasitas kekuatan *ultimate* sambungan menunjukkan hubungan yang konsisten secara non-linear.





# **TUGAS AKHIR**

**KATA PENGANTAR**



## KATA PENGANTAR

Puji syukur kehadirat ALLAH SWT. atas rahmat dan karunia-Nya hingga dapat terselesaikannya tugas akhir ini dengan judul :

### **ANALISA KAPASITAS KEKUATAN ULTIMATE DARI SAMBUNGAN BEAM DAN PELAT PADA TUBULAR MEMBER**

Adapun tugas akhir dengan beban 6 sks ini disusun guna memenuhi salah satu persyaratan dalam menyelesaikan program sarjana di Program Studi Teknik Kelautan Fakultas Teknologi Kelautan ITS.

Bimbingan dan simpati banyak penulis terima dari semua pihak selama penyusunan tugas akhir, dan dengan telah terselesaikannya tugas akhir ini penulis secara khusus mengucapkan terima kasih kepada :

1. Bapak DR. Ir. Eko Budi Djatniko, Msc. selaku Ketua Program Studi Teknik Kelautan FTK ITS.
2. Bapak DR. Ir. D. M. Rosyid selaku Dosen Wali sekaligus sebagai Dosen Pembimbing, atas petunjuk dan saran selama menempuh perkuliahan hingga penyusunan tugas akhir.
3. Pimpinan dan seluruh staf karyawan Perpustakaan ITS.
4. Rekan Choliq dan Agus (L-89), atas penyediaan paper dan copy master program.
5. Seluruh rekan angkatan 90 dan mahasiswa Teknik Kelautan FTK ITS atas simpati dan dukungannya.

Semoga Tuhan Yang Pemurah memberi balasan atas segala kerelaannya.

Akhir kata tugas akhir ini disusun dengan segenap kemampuan yang ada dengan harapan dapat dihasilkan suatu karya yang sebaik mungkin, namun terasa bahwa hasilnya belumlah sempurna. Untuk itu penulis berharap adanya pemikiran lebih lanjut dari pembaca sehingga buku tugas akhir ini dapat lebih bermanfaat.

penulis





# **TUGAS AKHIR**

**DAFTAR ISI**



## DAFTAR ISI

	halaman
JUDUL .....	i
PENGESAHAN .....	ii
SURAT KEPUTUSAN TUGAS AKHIR .....	iii
ABSTRAK .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xii
 BAB I PENDAHULUAN .....	 1
1.1. Latar Belakang Masalah .....	1
1.2. Perumusan Masalah .....	4
1.3. Tujuan Penulisan .....	5
1.4. Batasan Masalah .....	5
1.5. Metode Penyelesaian .....	6
1.6. Kerangka Bahasan .....	7
 BAB II FORMULASI ELEMEN SHELL ISOPARAMETRIS .....	 9
2.1. Pendahuluan .....	9
2.2. Sistem Koordinat .....	10
2.3. Geometri Elemen .....	13
2.4. Displasmen .....	14
2.5. Regangan dan Tegangan .....	16
2.6. Matrik Kekakuan .....	18
2.7. Integrasi Numerik .....	18



<b>BAB III</b>	<b>PENYELESAIAN PERSAMAAN ALJABAR LINEAR</b>	
	<b>SIMULTAN</b>	21
3.1.	Pendahuluan	21
3.2.	Metode Frontal	22
3.3.	Eliminasi Gauss	23
3.4.	Proses Kondensasi Statis	27
<b>BAB IV</b>	<b>PERMASALAHAN NON-LINEAR PADA METODE</b>	
	<b>ELEMEN HINGGA</b>	29
4.1	Pendahuluan	29
4.2.	Metode Penyelesaian pada Permasalahan Non-linear	30
4.2.1.	Substitusi Langsung	31
4.2.2.	Metode Newton Raphson (N-R)	32
4.2.3.	Metode Incremental	34
4.3.	Permasalahan Elastis-Plastis	35
4.4.	Hubungan Tegangan-Regangan pada Plastisitas	39
4.5.	Prosedur Analisa Elastis-Plastis	42
<b>BAB V</b>	<b>APLIKASI PROGRAM</b>	45
5.1.	Pendahuluan	45
5.2.	Program Utama	45
5.3.	Program Subroutine	47
5.4.	Variabel Input Data	59
5.5.	Variabel-variabel Program	64
5.6.	Program Analisa Grafis	65
<b>BAB VI</b>	<b>ANALISA HASIL PROGRAM</b>	66
6.1.	Analisa Hasil dan Pembahasan	66
6.1.1.	Pemodelan Elemen	66
6.1.2.	Data Geometri Elemen	72

6.1.3. Data Material .....	73
6.1.4. Kondisi Pembebanan .....	73
6.1.5. Perbandingan Hasil Nilai Kapasitas Ultimate .....	77
6.1.6 Analisa Pengaruh Perubahan Parameter $\beta$ .....	78
6.1.7. Hubungan Tegangan dan Deformasi .....	79
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>88</b>
<b>DAFTAR PUSTAKA .....</b>	<b>90</b>
<b>LAMPIRAN 1. LISTING PROGRAM UTAMA .....</b>	<b>91</b>
<b>LAMPIRAN 2. LISTING PROGRAM PLOT GRAFIS .....</b>	<b>128</b>
<b>LAMPIRAN 3. CONTOH INPUT DATA PROGRAM .....</b>	<b>134</b>
<b>LAMPIRAN 4. CONTOH OUTPUT DATA KONTROL ELEMEN DAN DISPLASMEN .....</b>	<b>139</b>
<b>LAMPIRAN 5. CONTOH OUTPUT DATA TEGANGAN DAN RESULTAN GAYA .....</b>	<b>153</b>





# **TUGAS AKHIR**

## **DAFTAR GAMBAR**



## D A F T A R G A M B A R

	halaman
Gambar 1.1. Letak model sambungan IT dan PT <i>joint</i> pada struktur truss geladak .....	3
Gambar 1.2. Model sambungan PT dan IT <i>joint</i> .....	3
Gambar 1.3. <i>Flow chart</i> prosedur analisa P <i>ultimate</i> .....	6
Gambar 1.4. Lanjutan <i>flow chart</i> prosedur analisa P <i>ultimate</i> .....	7
Gambar 2.1. Sistem koordinat global .....	10
Gambar 2.2. Sistem koordinat simpul .....	11
Gambar 2.3. Displasmen pada suatu nodal poin.....	15
Gambar 2.4. Posisi Gauss point .....	20
Gambar 3.1. Idealisasi penyelesaian frontal .....	22
Gambar 3.2. Sistem elemen truss .....	27
Gambar 4.1. Pegas non-linear .....	30
Gambar 4.2. Interpretasi grafik dari pers. 4.2-2 .....	32
Gambar 4.3. Interpretasi grafik dari pers. 4.2-3 .....	32
Gambar 4.4. Interpretasi grafik pers. N-R .....	33
Gambar 4.5. Interpretasi grafik pers. 4.2-13 .....	35
Gambar 4.6. Interpretasi grafik pers. 4.2-14 .....	35
Gambar 4.7. Hubungan tegangan-regangan .....	36
Gambar 4.8. Struktur bar dengan model finite elemennya.....	38
Gambar 4.9. Proses penyelesaian metode tangent-stiffness .....	38
Gambar 4.10. Hubungan tegangan regangan pada kondisi yield .....	39
Gambar 5.1. <i>Flow chart</i> program utama .....	46
Gambar 5.2. Lanjutan <i>flow chart</i> program utama .....	47



Gambar 5.3.	<i>Flow chart</i> Subroutine BGMAT .....	48
Gambar 5.4.	Lanjutan <i>flow chart</i> Subroutine BGMAT .....	49
Gambar 5.5.	<i>Flow chart</i> Subroutine CONVER .....	52
Gambar 5.6.	<i>Flow chart</i> Subroutine FRONT .....	53
Gambar 5.7.	Lanjutan <i>flow</i> Subroutine FRONT .....	54
Gambar 5.8.	<i>Flow chart</i> Subroutine RESTR .....	56
Gambar 5.9.	Lanjutan <i>flow</i> Subroutine RESTR .....	57
Gambar 5.10.	<i>Flow chart</i> Subroutine STIFF .....	58
Gambar 6.1.	Model elemen IT <i>joint</i> .....	68
Gambar 6.2.	Model elemen PT <i>joint</i> .....	69
Gambar 6.3.	Pemilihan jenis integrasi <i>reduced</i> (R), <i>selective</i> (S) dan <i>full</i> (F) menurut jenis elemen .....	70
Gambar 6.4.	Bentuk elemen yang tidak baik .....	71
Gambar 6.5.	Bentuk hubungan nodal point antar elemen yang tidak baik ....	71
Gambar 6.6.	Grafik Estimasi Kapasitas P <i>Ultimate</i> .....	79
Gambar 6.7.	Grafik Hubungan Tegangan-Displasmen (IT <i>joint</i> , $\beta=0.5$ ) ....	80
Gambar 6.8.	Grafik Hubungan Tegangan-Displasmen (IT <i>joint</i> , $\beta=0.6$ ) ....	80
Gambar 6.9.	Grafik Hubungan Tegangan-Displasmen (IT <i>joint</i> , $\beta=0.7$ ) ....	81
Gambar 6.10.	Grafik Hubungan Tegangan-Displasmen (IT <i>joint</i> , $\beta=0.8$ ) ....	81
Gambar 6.11.	Grafik Hubungan Tegangan-Displasmen (IT <i>joint</i> , $\beta=0.9$ ) ....	82
Gambar 6.12.	Grafik Hubungan Tegangan-Displasmen (PT <i>joint</i> , $\beta=0.5$ ) ....	82
Gambar 6.13.	Grafik Hubungan Tegangan-Displasmen (PT <i>joint</i> , $\beta=0.6$ ) ....	83
Gambar 6.14.	Grafik Hubungan Tegangan-Displasmen (PT <i>joint</i> , $\beta=0.7$ ) ....	83
Gambar 6.15.	Grafik Hubungan Tegangan-Displasmen (PT <i>joint</i> , $\beta=0.8$ ) ....	84
Gambar 6.16.	Grafik Hubungan Tegangan-Displasmen (PT <i>joint</i> , $\beta=0.9$ ) ....	84
Gambar 6.17.	Penyelesaian konvergensi sebelum dan sesudah P <i>ultimate</i> ....	85
Gambar 6.18.	Deformasi tubular setelah Pult .....	87





# **TUGAS AKHIR**

**DAFTAR TABEL**



## DAFTAR TABEL

	halaman
Tabel 6.1. Data geometri model elemen .....	72
Tabel 6.2. <i>P ultimate</i> dari hasil pengujian, penurunan rumus dan hasil analisa program .....	77
Tabel 6.3. Perbandingan <i>P ultimate</i> pengujian dengan hasil <i>P ultimate</i> penurunan rumus dan hasil analisa program.....	78
Tabel 6.4. Estimasi <i>P ultimate</i> dengan perubahan parameter $\beta$ [kN] .....	78



# **TUGAS AKHIR**

## **BAB I**

### **PENDAHULUAN**



## BAB I PENDAHULUAN



### 1.1. Latar Belakang Masalah

Komponen-komponen struktur yang menyusun struktur platform bangunan lepas pantai secara prinsipial terdiri dari tiga bagian yaitu struktur jacket, pile dan struktur geladak. Struktur geladak mempunyai fungsi selain sebagai penyedia area tempat aktivitas (*working area*) juga sebagai pendukung bagi seluruh peralatan maupun bangunan-bangunan modul yang ada diatas struktur jacket. Struktur geladak juga diperkuat dengan komponen-komponen struktur (*members*) yang dipersatukan menjadi satu bagian dari struktur truss geladak. Kekuatan dari struktur tersebut dalam menahan beban tergantung pada ukuran dan geometri dari tiap bagian sambungannya.

Pada umumnya struktur truss geladak terdiri dari sambungan yang merupakan sambungan antara *beam*, pelat dan tubular dimana bentuk sambungan tersebut dapat terdiri dari sambungan antara *beam* (I-*beam*) atau pelat yang menyatu pada tubular (IT/PT *joint*) dan sambungan antara tubular yang menyatu pada *beam* (TI *joint*). Bagian-bagian sambungan tersebut dapat dijumpai pada bagian sambungan antara kaki geladak dengan pelat/*beam* geladak seperti yang tampak pada Gambar 1.1.

Analisa kekuatan sambungan struktur truss geladak dilakukan untuk mendapatkan ukuran dan geometri sambungan yang sesuai. Analisa statis dapat digunakan untuk menganalisa respon sambungan terhadap pembebanan yang ada sehingga dapat diketahui kapasitas kekuatan *ultimate* sambungan dalam menerima beban. Penyelidikan untuk mendapatkan persamaan kapasitas kekuatan *ultimate* dari tipe-tipe sambungan dengan berbagai jenis pembebanannya pada permasalahan

di atas telah dilakukan oleh beberapa orang diantaranya oleh Kamba, Kurobane, Wardenier dan Steinmetz [Ref 1]. Peneliti-peneliti tersebut telah berhasil memperoleh persamaan-persamaan kekuatan *ultimate* yang berbeda-beda.

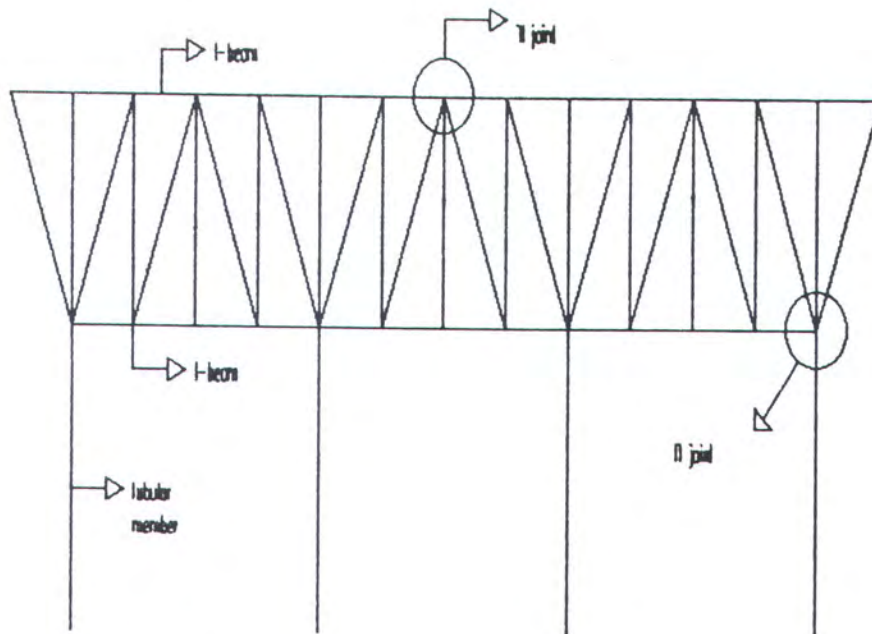
Adanya perbedaan diantara persamaan-persamaan tersebut adalah karena perbedaan dalam meninjau batas kriteria kapasitas kekuatan *ultimate* dari sambungan. Kriteria untuk menentukan kapasitas kekuatan sambungan tersebut antara lain, pertama adalah seperti beban maksimum yang dicapai selama proses pembebanan hingga struktur sambungan tersebut mengalami cacat atau retak pertama kalinya. Kedua, untuk penentuan kriterianya adalah ketika beban sudah menyebabkan deformasi yang berlebihan pada struktur sambungan hingga melebihi dari batas lendutan yang telah ditetapkan. Selain itu penurunan persamaan baru yang ada juga merupakan hasil perbandingan dan koreksi dari persamaan-persamaan sebelumnya sehingga penurunan persamaan yang baru tersebut lebih konsisten terhadap hasil yang akan diprediksi dan juga persamaannya menjadi lebih sederhana.

Pada dasarnya tujuan dari penurunan persamaan kapasitas kekuatan *ultimate* sambungan adalah untuk mengetahui nilai kapasitas kekuatan *ultimate* sambungan akibat pengaruh perubahan parameter-parameter geometri sambungan. Parameter-parameter geometri sambungan tersebut adalah sebagai berikut :

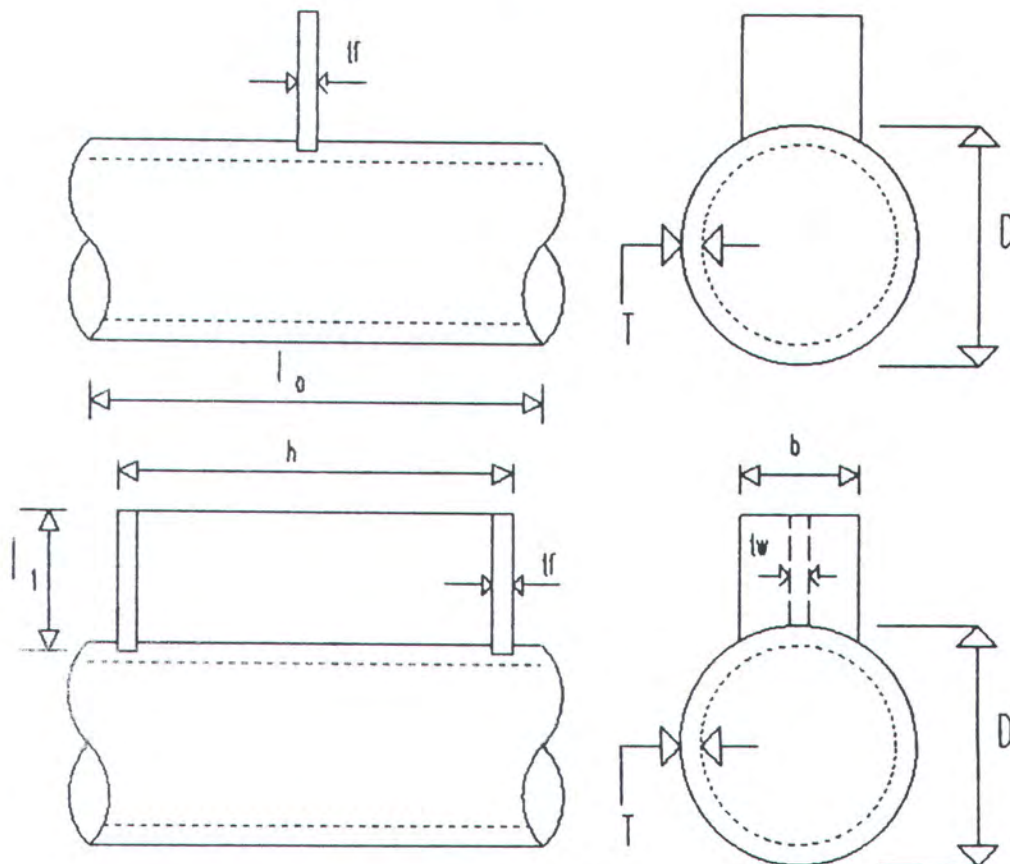
- $\gamma$ , perbandingan antara diameter chord (tubular) dengan ketebalannya ( $D/2T$ ).
- $\beta$ , perbandingan antara lebar *beam* atau pelat dengan diameter chord ( $b/D$ ).
- $\eta$ , perbandingan antara tinggi *beam* atau pelat dengan diameter chord ( $h/D$ ).

Macam konfigurasi pembebanan yang secara umum dikenakan pada permasalahan diatas adalah beban aksial (tarik atau tekan) dan beban moment ataupun kombinasi dari keduanya. Penjelasan parameter geometri sambungan adalah seperti yang terlihat pada Gambar 1.2.





Gambar 1.1 Letak model sambungan IT dan TI *joint* pada struktur truss geladak



Gambar 1.2. Model sambungan PT dan IT *joint*

Keterbatasan dalam hal besarnya jumlah *range* ukuran geometri dari tipe sambungan yang harus digunakan maupun macam konfigurasi dari jenis pembebanan yang akan dikenakan dalam pengujian untuk mendapatkan penurunan persamaan kapasitas *ultimate* sambungan, dapat menyebabkan kesulitan dalam memprediksi nilai kapasitas kekuatan *ultimate* sambungan secara akurat.

Latar belakang permasalahan di atas menyebabkan penulis untuk mencoba melakukan analisa secara numerik dengan pendekatan metode elemen hingga dalam memprediksi kapasitas kekuatan *ultimate* sambungan seperti pada permasalahan. Analisa ini juga dilatarbelakangi oleh adanya perkembangan metode atau teori pendekatan dalam menyelesaikan permasalahan mekanika struktur pada metode elemen hingga khususnya dalam hal ini adalah permasalahan menentukan kapasitas kekuatan *ultimate* sambungan. Disamping itu dengan adanya perkembangan dunia komputer maka penyelesaian secara numerik tersebut dapat dilakukan dengan cepat dan efisien.

Dalam tugas akhir ini penyusunan program khusus dilakukan untuk menganalisa penyelesaian masalah kapasitas kekuatan *ultimate* sambungan dengan memperhatikan lingkup batasan permasalahan yang akan dianalisa. Pada akhirnya hasil dari program khusus ini diharapkan dapat *memberikan* informasi-informasi secara jelas tentang keadaan deformasi maupun kondisi tegangan elemen. Informasi-informasi tersebut dengan berdasarkan teori maupun metode yang ada, dapat digunakan untuk menganalisa dan memprediksi kapasitas kekuatan *ultimate* sambungan.

## 1.2. Perumusan Masalah

Perumusan masalah pada tugas akhir ini adalah merupakan pertanyaan-pertanyaan atau masalah yang timbul pada saat analisa kapasitas kekuatan *ultimate* sambungan seperti :



- Pendekatan analisa numerik yang bagaimanakah pada konsep metode elemen hingga yang dapat digunakan dalam menganalisa kapasitas kekuatan *ultimate* sambungan.
- Apakah yang dapat disimpulkan dari perbandingan antara analisa kapasitas kekuatan *ultimate* sambungan melalui analisa pendekatan numerik metode elemen hingga dengan hasil data pengujian maupun penurunan rumus kapasitas kekuatan *ultimate* sambungan yang ada.
- Bagaimanakah pengaruh parameter geometri sambungan dalam hal ini adalah parameter  $\beta$  terhadap kapasitas kekuatan *ultimate* sambungan.

### 1.3 Tujuan Penulisan

Tujuan penulisan tugas akhir ini adalah untuk menganalisa kapasitas kekuatan *ultimate* sambungan antara *I-beam* dan pelat pada tubular (IT dan PT *joint*) dengan kondisi pembebanan aksial dan selain itu juga menganalisa pengaruh perubahan parameter geometri terhadap kapasitas kekuatan sambungan yang dilakukan dengan analisa pendekatan metode elemen hingga.

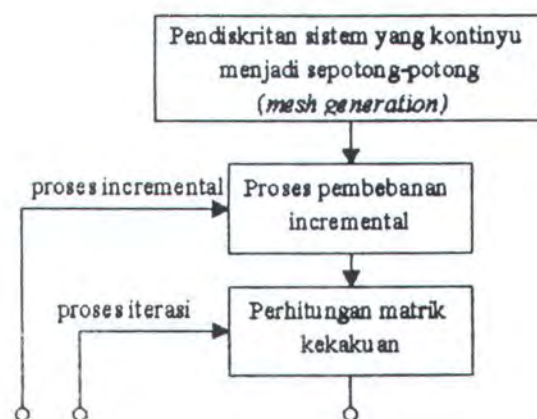
### 1.4. Batasan Masalah

Batasan masalah pada tugas akhir ini dipakai untuk mempermudah analisa yaitu seperti :

- Tipe sambungan yang akan dianalisa adalah IT dan PT *joint*.
- Bentuk pemodelan elemen sambungan adalah elemen *shell*.
- Jenis pembebanan adalah beban statis aksial tekan.
- Pengaruh parameter geometri sambungan yang ditinjau hanya parameter  $\beta$ .
- Pada sambungan dianggap tidak terdapat stiffeners.
- Jenis tumpuan dianggap pin pada kedua ujung chord.
- Analisa pada daerah lokal *weld toe* diabaikan.

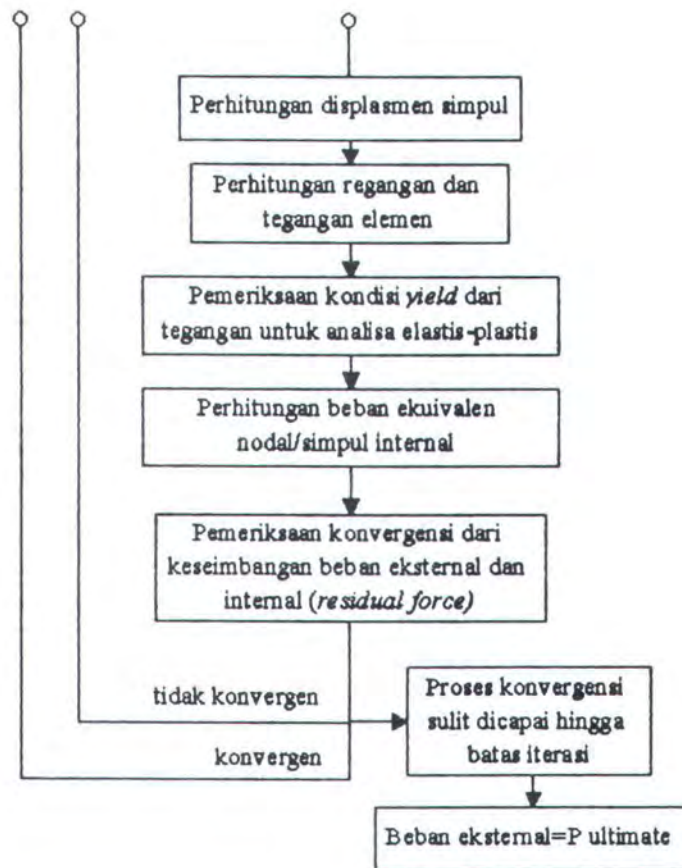
### 1.5. Metode Penyelesaian

Metode penyelesaian yang digunakan dalam analisa kapasitas kekuatan *ultimate* sambungan *beam* dan pelat pada tubular member (IT dan PT *joint*) pada tugas akhir ini adalah dengan penyusunan suatu program khusus dalam bahasa FORTRAN yang merupakan hasil modifikasi dari referensi program tertulis yang disusun oleh J. A. FIGUERAS dan D. R. J. OWEN [Ref. 2]. Program tersebut pada dasarnya merupakan pemanfaatan beberapa formulasi dasar dan pengembangan teori analisa statis nonlinear dalam lingkup konsep pendekatan dengan metode elemen hingga. Secara umum sistem sambungan yang akan dianalisa adalah dianggap sebagai gabungan dari sejumlah elemen kecil yang dihubungkan dengan titik-titik simpul dalam hal ini model elemen yang dipakai adalah elemen *shell* dan dengan mempertimbangkan kompatibilitas antar elemen pada tiap simpul maka gabungan dari elemen-elemen tersebut akan dapat menggambarkan karakteristik dari sistem yang akan dianalisa. Sedangkan garis besar prosedur analisa yang akan dilakukan adalah sebagai berikut :



Gambar 1.3. Flow chart prosedur analisa *P ultimate*.





Gambar 1.4. Lanjutan *flow chart* prosedur analisa *P ultimate*.

## 1.6. Kerangka Bahasan

Pembahasan analisa kapasitas kekuatan ultimate dari sambungan *beam* dan pelat pada tubular member (IT dan PT joint) di dalam tugas akhir akan diuraikan menjadi beberapa bab setelah bab pendahuluan ini. Pembahasan pada bab-bab berikut akan meliputi antara lain ; penjelasan formulasi dasar metode elemen hingga yang digunakan dalam perhitungan, pembahasan permasalahan non-linear, penjelasan aplikasi program hingga analisa hasil program dan kesimpulan. Secara garis besar pembahasan yang ada dapat diuraikan dalam kerangka bahasan berikut ini :

1. Formulasi elemen *shell* isoparametris, yang membahas :
  - Pendefinisian sistem koordinat yang digunakan sebagai sistem geometri dari model elemen.
  - Kondisi displasmen.
  - Hubungan tegangan dan regangan.
  - Penurunan matrik kekakuan.
  - Metode perhitungan integrasi.
2. Penyelesaian persamaan aljabar simultan, yang membahas :
  - Konsep pembentukan persamaan aljabar simultan dengan metode *wave front*.
  - Proses penyelesaian matrik persamaan dengan metode eliminasi *Gauss*.
3. Permasalahan non-linear pada metode elemen hingga, yang membahas :
  - Metode penyelesaian permasalahan non-linear.
  - Permasalahan elastis-plastis.
  - Hubungan tegangan-regangan pada plastisitas.
  - Prosedur analisa elastis plastis.
4. Aplikasi program, yang membahas :
  - Penjelasan program utama dan subroutine program.
  - Penjelasan variabel-variabel program.
5. Analisa hasil program, yang membahas :
  - Kondisi pemodelan dan pembebanan pada elemen.
  - Hasil perbandingan nilai kapasitas *ultimate* antara penyelesaian program dengan data referensi hasil pengujian dan penurunan rumus.
  - Pengaruh perubahan parameter geometri  $\beta$  terhadap nilai kapasitas *ultimate*.
  - Hubungan tegangan dan deformasi pada saat pembebanan.
6. Kesimpulan.





# **TUGAS AKHIR**

## **BAB II**

### **FORMULASI ELEMEN SHELL ISOPARAMETRIS**



## BAB II

### FORMULASI ELEMEN SHELL ISOPARAMETRIS

#### 2.1. Pendahuluan

Formulasi isoparametris sangat memungkinkan untuk digunakan pada generasi elemen yang mempunyai sisi kurva atau lengkung. Bentuk model elemen yang sedemikian rupa tersebut tentunya dimaksudkan untuk mendekati dengan bentuk model struktur yang akan dianalisa. Formulasi ini dapat digunakan pada permasalahan elemen plane, pejal maupun *shell*.

Dalam formulasinya, elemen isoparametris menggunakan sistem koordinat natural yaitu  $\xi$ ,  $\eta$  dan  $\zeta$  sehingga displasmen yang terjadi diekspresikan juga dalam bentuk koordinat natural dan untuk transformasi ke bentuk koordinat global  $x$ ,  $y$  dan  $z$  diperlukan suatu matrik transformasi Jacobian  $[J]$ .

Elemen isoparametris mempunyai arti bahwa parameter elemen yang digunakan untuk mengekspresikan bentuk geometri dari elemen adalah sama (iso) jumlahnya dengan parameter yang digunakan untuk mengekspresikan model displasmen. Hal ini dapat dijelaskan seperti berikut :

- Nodal d.o.f.  $\{d\}$  yang digunakan untuk mendefinisikan displasmen  $[u \ v \ w]$  pada node elemen adalah  $[u \ v \ w]^T = [N] \{d\}$
- Nodal koordinat  $\{c\}$  yang digunakan untuk mendefinisikan koordinat  $[x \ y \ z]$  pada titik simpul elemen adalah  $[x \ y \ z]^T = [N]' \{c\}$

dimana  $[N]$  dan  $[N]'$  adalah matrik *shape function* dalam fungsi koordinat natural  $\xi$ ,  $\eta$  dan  $\zeta$ . Elemen tersebut disebut elemen isoparametris jika parameter  $[N]$  dan  $[N]'$  adalah identik atau sama dan jika jumlah parameter  $[N]$  adalah lebih sedikit



daripada jumlah parameter  $[N]'$  maka elemen dapat disebut sebagai superparametris demikian juga sebaliknya untuk elemen subparametris.

## 2.2. Sistem Koordinat

Dalam formulasi degenerasi kurva elemen *shell*, sistem koordinat ditentukan secara khusus. Terdapat empat sistem koordinat yang dipakai dalam formulasi, yaitu sebagai berikut :

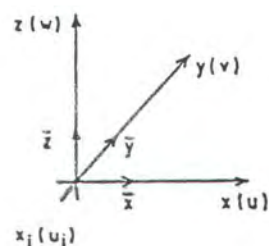
### 1. Sistem koordinat global $x_i$

Sistem koordinat global dipilih secara bebas untuk menggambarkan bentuk geometri elemen *shell* seutuhnya seperti terlihat pada Gambar 2.1. Koordinat titik simpul, displasmen dan vektor gaya eksternal adalah menggunakan referensi sistem koordinat ini. Adapun notasi yang digunakan dalam sistem koordinat global adalah sebagai berikut :

$$x_{i(i=1,3)} \text{ dimana } x_1 = x, x_2 = y, x_3 = z \quad (2.2-1)$$

$$u_{i(i=1,3)} \text{ dimana } u_1 = u, u_2 = v, u_3 = w \quad (2.2-2)$$

$$\bar{x}_{i(i=1,3)} \text{ adalah unit vektor pada arah } x_i \quad (2.2-3)$$



Gambar 2.1. Sistem koordinat global.

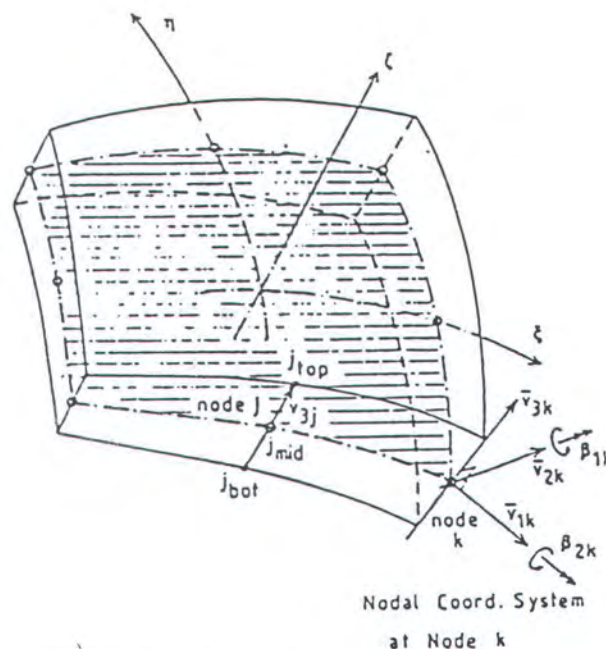
## 2. Sistem koordinat simpul $\underline{v}_{ik}$

Sistem koordinat simpul ditentukan pada masing-masing titik simpul dari elemen dengan menggunakan referensi *mid-surface* dari elemen sebelum maupun sesudah mengalami deformasi, hal ini seperti yang dijelaskan pada Gambar 2.2. Pada gambar untuk titik simpul k dapat ditulis suatu vektor dengan arah ketebalan  $\underline{v}_{3k}$  yang besarnya ditentukan oleh selisih koordinat titik pada permukaan atas dan bawah dari elemen pada titik simpul k.

$$\underline{v}_{3k} = \underline{x}_k^{\text{top}} - \underline{x}_k^{\text{bot}} \quad (2.2-4)$$

dimana,  $\underline{x}_k = [x_k \ y_k \ z_k]^T$  sehingga persamaan selanjutnya dapat ditulis dalam bentuk matrik seperti berikut :

$$\underline{v}_{3k} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}^{\text{top}} - \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}^{\text{bot}} \quad (2.2-5)$$



Gambar 2.2. Sistem koordinat simpul.



sedangkan vektor  $\underline{v}_{1k}$  dan  $\underline{v}_{2k}$  seperti yang terlihat pada gambar adalah tegak lurus dengan vektor  $\underline{v}_{3k}$  dan mempunyai fungsi untuk mendefinisikan besarnya rotasi pada simpul k. Jika dalam kondisi sistem koordinat global, vektor  $\underline{v}_{1k}$  dan  $\underline{v}_{3k}$  adalah paralel pada satu bidang dan vektor  $\underline{v}_{2k}$  tegak lurus dengan bidang tersebut maka hubungan dari ketiga vektor tersebut dapat ditulis :

$$\underline{v}_{2k} = \underline{v}_{3k} \times \underline{v}_{1k} \quad (2.2-6)$$

### 3. Sistem koordinat natural $\xi$ , $\eta$ dan $\zeta$

Pada sistem koordinat ini,  $\xi$  dan  $\eta$  adalah dua koordinat kurvalinear pada bidang tengah dari elemen sedangkan  $\zeta$  adalah koordinat linear pada arah ketebalan elemen dimana arah  $\zeta$  tegak lurus dengan bidang *mid-surface* dari elemen *shell* dan juga merupakan fungsi dari unit vektor  $\underline{v}_{3k}$ . Dan masing-masing koordinat diasumsikan mempunyai harga maksimum +1 dan harga minimum -1.

### 4. Sistem koordinat lokal $x'_i$

Sistem koordinat lokal ditentukan pada masing-masing titik simpul dimana tegangan dan regangan akan dihitung. Karena arah dari koordinat  $x'_3$  ( $z'$ ) adalah tegak lurus dengan permukaan  $\zeta$  yang konstan, maka besar dari  $x'_3$  ( $z'$ ) merupakan hasil dari perkalian dari matrik :

$$\underline{x}_3 = \begin{bmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \\ \frac{\partial z}{\partial \xi} \end{bmatrix} \times \begin{bmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \eta} \end{bmatrix} \quad (2.2-7)$$

dan untuk arah dari koordinat  $\underline{x}_1'$  ( $x'$ ) dapat dianggap sebagai persinggungan pada arah  $\xi$  sehingga besar  $\underline{x}_1'$  ( $x'$ ) dapat diekspresikan dengan persamaan matrik berikut :

$$\underline{x}_1' = \begin{bmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \xi} \end{bmatrix} \quad (2.2-8)$$

sedangkan untuk arah koordinat  $\underline{x}_2'$  ( $y'$ ) didefinisikan sebagai hasil dari perkalian :

$$\underline{x}_2' = \underline{x}_3' \times \underline{x}_1' \quad (2.2-9)$$

Hubungan antara sistem koordinat lokal dengan koordinat global dinyatakan dengan suatu matrik kosinus  $[\theta]$  yang didefinisikan dengan persamaan berikut :

$$[\theta] = [\bar{x}' \quad \bar{y}' \quad \bar{z}'] \quad (2.2-10)$$

dimana  $\bar{x}'$ ,  $\bar{y}'$ ,  $\bar{z}'$  adalah unit vektor pada arah koordinat lokal  $x'$ ,  $y'$  dan  $z'$ .

### 2.3. Geometri Elemen

Koordinat global dari suatu titik simpul pada titik permukaan atas (*top coordinate*) dan titik permukaan bawah (*bottom coordinate*) pada suatu elemen biasa digunakan sebagai input geometri elemen. Sebagai alternatif dari input geometri elemen dapat juga digunakan koordinat titik permukaan tengah-tengah (*mid-surface coordinate*) pada suatu elemen yang dihubungkan dengan unit vektor arah ketebalan  $\bar{v}_{3k}$ . Pada formulasi isoparametris, koordinat titik simpul dalam



suatu elemen diperoleh dengan menerapkan suatu *shape function* elemen dan perumusannya adalah sebagai berikut :

$$x_i = \sum_{k=1}^n N_k \frac{1+\zeta}{2} x_{ik}^{top} + \sum_{k=1}^n N_k \frac{1-\zeta}{2} x_{ik}^{bot} \quad (2.3-1)$$

atau,

$$x_i = \sum_{k=1}^n N_k x_{ik}^{mid} + \sum_{k=1}^n N_k \frac{h_k}{2} \zeta \frac{-i}{V_{3k}} \quad (2.3-2)$$

dimana :

$i = 1, 3$  arah koordinat global

$n =$  jumlah titik simpul per elemen

$N_k =$  *shape function* yang merupakan fungsi dari  $\zeta$  dan  $\eta$  pada permukaan  $\zeta$  yang konstan.

$k = 1 \dots n$

$h_k =$  ketebalan elemen *shell*

Pada persamaan 2.3-2 dalam bentuk matrik dapat ditulis :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \sum_{k=1}^n N_k \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}_{mid} + \sum_{k=1}^n N_k \frac{h_k}{2} \zeta \begin{bmatrix} -x \\ V_{3k} \\ -y \\ V_{3k} \\ -z \\ V_{3k} \end{bmatrix} \quad (2.3-3)$$

## 2.4. Displasmen

Pada analisa elemen *shell* diasumsikan terdapat lima derajat kebebasan, yaitu tiga gerakan translasi yang terjadi di titik permukaan tengah (*mid point*) dari elemen ( $u_{ik}^{mid}$ ) dan dua gerakan rotasi ( $\beta_{1k}, \beta_{2k}$ ) seperti yang terlihat pada Gambar 2.3. Komponen displasmen dari titik simpul elemen tersebut dapat ditulis dengan persamaan :

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \sum_{k=1}^n N_k \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix}_{mid} + \sum_{k=1}^n N_k \zeta \frac{h_k}{2} \begin{bmatrix} v_{1k}^x & v_{2k}^x \\ v_{1k}^y & v_{2k}^y \\ v_{1k}^z & v_{2k}^z \end{bmatrix} \begin{bmatrix} \beta_{1k} \\ \beta_{2k} \end{bmatrix} \quad (2.4-1)$$

atau dapat ditulis :

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} N_k & 0 & 0 & N_k \zeta \frac{h_k}{2} v_{1k}^x & -N_k \zeta \frac{h_k}{2} v_{2k}^x \\ 0 & N_k & 0 & N_k \zeta \frac{h_k}{2} v_{1k}^y & -N_k \zeta \frac{h_k}{2} v_{2k}^y \\ 0 & 0 & N_k & N_k \zeta \frac{h_k}{2} v_{1k}^z & -N_k \zeta \frac{h_k}{2} v_{2k}^z \end{bmatrix} \begin{bmatrix} u_k \\ v_k \\ w_k \\ \beta_{1k} \\ \beta_{2k} \end{bmatrix} \quad (2.4-2)$$

dalam bentuk persamaan umum dapat ditulis sebagai :

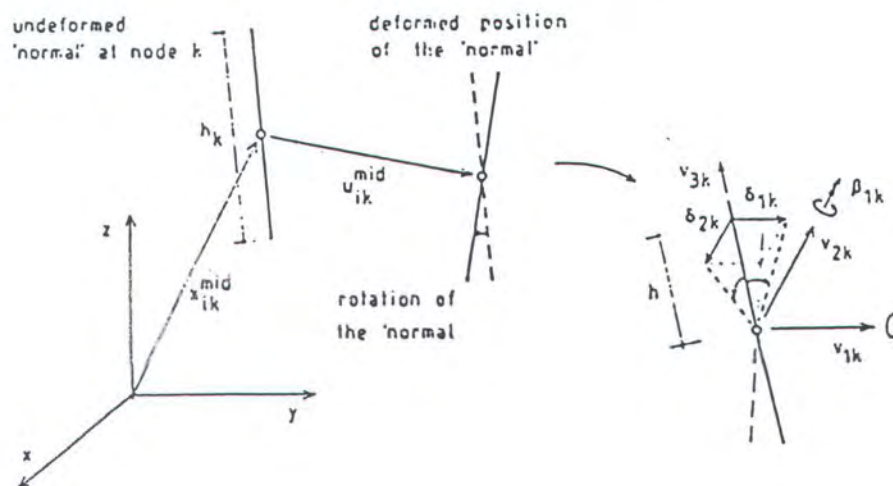
$$u_k = N_k \delta_k \quad (2.4-3)$$

untuk keseluruhan elemen maka persamaan displasmen menjadi :

$$u = N \delta \quad (2.4-4)$$

dimana  $N$  = matrik *shape function*

$\delta$  = variabel vektor displasmen dari titik simpul elemen



Gambar 2.3. Displasmen pada suatu nodal poin.



## 2.5. Regangan dan Tegangan

Pada analisa elemen *shell*, diasumsikan bahwa tegangan normal pada arah  $z'$  adalah nol ( $\sigma_{z'} = 0$ ) dimana dalam hal ini  $z'$  adalah merupakan sistem koordinat lokal  $x_i$  dan tegak lurus dengan bidang  $\xi\eta$ , sedangkan komponen-komponen regangan pada elemen *shell* adalah :

$$\varepsilon = \begin{bmatrix} \varepsilon_{x'} \\ \varepsilon_{y'} \\ \gamma_{x'y'} \\ \gamma_{x'z'} \\ \gamma_{y'z'} \end{bmatrix} = \begin{bmatrix} \partial u' / \partial x' \\ \partial v' / \partial y' \\ \partial u' / \partial y' + \partial v' / \partial x' \\ \partial u' / \partial z' + \partial w' / \partial x' \\ \partial v' / \partial z' + \partial w' / \partial y' \end{bmatrix} \quad (2.5-1)$$

dimana  $u'$ ,  $v'$ ,  $w'$  adalah komponen-komponen dispasmen pada koordinat lokal. Persamaan derivatif dispasmen lokal yang terjadi diperoleh dari transformasi persamaan derivatif dispasmen global  $u$ ,  $v$ ,  $w$  dimana hubungan persamaan tersebut dapat ditulis :

$$\begin{bmatrix} \frac{\partial u'}{\partial x'} & \frac{\partial v'}{\partial x'} & \frac{\partial w'}{\partial x'} \\ \frac{\partial u'}{\partial y'} & \frac{\partial v'}{\partial y'} & \frac{\partial w'}{\partial y'} \\ \frac{\partial u'}{\partial z'} & \frac{\partial v'}{\partial z'} & \frac{\partial w'}{\partial z'} \end{bmatrix} = [\theta]^T \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} [\theta] \quad (2.5-2)$$

dimana  $[\theta]$  adalah matrik transformasi, sedangkan dalam hal ini persamaan derivatif dispasmen global adalah merupakan hasil transformasi dari persamaan :

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{bmatrix} = [J]^{-1} \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial v}{\partial \xi} & \frac{\partial w}{\partial \xi} \\ \frac{\partial u}{\partial \eta} & \frac{\partial v}{\partial \eta} & \frac{\partial w}{\partial \eta} \\ \frac{\partial u}{\partial \zeta} & \frac{\partial v}{\partial \zeta} & \frac{\partial w}{\partial \zeta} \end{bmatrix} \quad (2.5-3)$$

dimana  $[J]^{-1}$  adalah matrik transformasi Jacobian yang dirumuskan sebagai berikut :

$$[J]^{-1} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (2.5-4)$$

Hubungan antara regangan dan displasmen yang terjadi pada permasalahan di atas dapat ditulis dengan persamaan :

$$\epsilon = B \delta \quad (2.5-5)$$

dimana B adalah matrik *strain-displasmen* yang mempunyai lima baris dan sejumlah kolom yang sesuai dengan jumlah titik simpul per elemen.

Permasalahan tegangan pada analisa elemen *shell* sebagaimana yang sebelumnya diasumsikan bahwa tegangan pada arah tegak lurus dengan permukaan tengah elemen (*mid surface*) adalah dianggap nol ( $\sigma_z = 0$ ) sehingga komponen-tegangan tegangan pada analisa elemen *shell* pada sistem koordinat lokal adalah :



$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{bmatrix} = E \epsilon \quad (2.5-6)$$

dimana E adalah matrik elastisitas yang komponennya merupakan sifat atau jenis material daripada elemen.

## 2.6. Matrik Kekakuan

Formulasi dengan metode energi regangan dan energi potensial, dan dengan diketahuinya matrik *strain-displasmen* [B] dapat digunakan untuk merumuskan matrik kekakuan sebagai berikut :

$$[K] = \iiint [B]^T [E] [B] dv \quad (2.6-1)$$

Dengan menggunakan referensi pada sistem koordinat natural maka persamaan di atas berubah menjadi :

$$[K] = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B]^T [E] [B] |J| d\xi d\eta d\zeta \quad (2.6-2)$$

dimana dalam hal ini matrik [B] merupakan fungsi dari  $\xi$ ,  $\eta$  dan  $\zeta$ .

## 2.7. Integrasi Numerik

Salah satu metode pendekatan yang paling sering digunakan pada perhitungan integral secara numerik seperti persamaan (2.6-2), khususnya pada analisa elemen isoparametris adalah metode *Gauss Quadrature*. Perhitungan integral ini dilakukan pada *sampling point* (*Gauss point*) dari elemen dengan

menggunakan konstanta integrasi *Gauss Quadrature* yang berupa *weight factor* ( $W$ ).

Fungsi persamaan yang akan diintegrasikan pada analisa elemen isoparametris adalah merupakan transformasi fungsi persamaan pada sistem global ( $f=f(x,y,z)$ ) menjadi fungsi persamaan pada sistem koordinat natural ( $\phi=\phi(\xi,\eta,\zeta)$ ) yang mempunyai batas integral antara -1 dan +1 dimana pada proses transformasi tersebut dilakukan oleh matrik transformasi Jacobian [J]. Secara umum hal di atas dapat ditulis dalam bentuk persamaan berikut :

$$I = \int_{x1}^{x2} \int_{y1}^{y2} \int_{z1}^{z2} f(x,y,z) dx dy dz \quad (2.7-1)$$

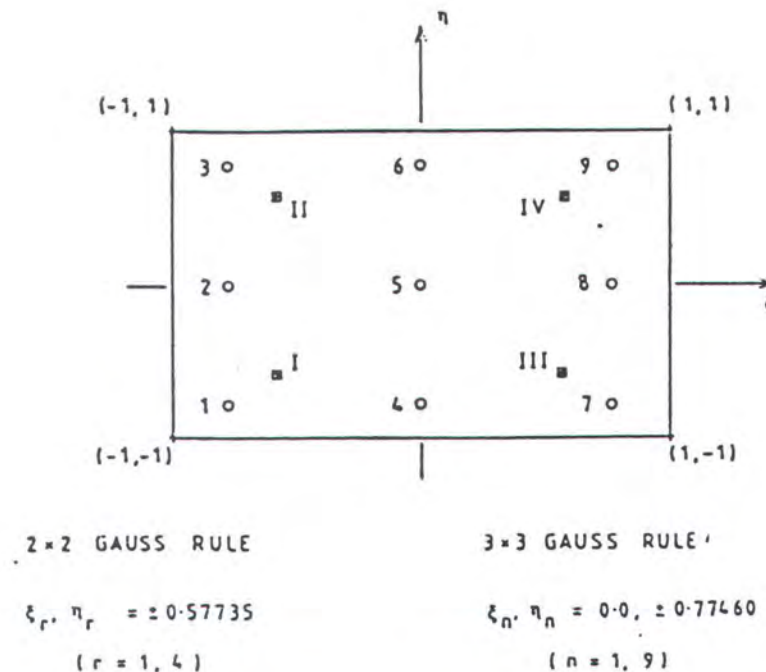
Persamaan di atas selanjutnya diubah menjadi :

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \phi(\xi,\eta,\zeta) d\xi d\eta d\zeta \quad (2.7-2)$$

Perhitungan pendekatan dari persamaan integral (2.7-2) dapat dilakukan dengan mengevaluasi fungsi  $\phi$  pada lokasi *Gauss point* tertentu pada elemen (hal ini seperti terlihat pada Gambar 2.4 untuk contoh elemen isoparametris dua dimensi) dan dalam perhitungannya, dari persamaan 2.7-2 dapat digeneralisasi menjadi formula quadrature yaitu dengan mengalikan masing-masing fungsi integrasi dengan konstanta *weight factor* ( $W$ ) dan kemudian menjumlahkannya. Hal tersebut dapat dijelaskan dengan persamaan sebagai berikut :

$$I = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \phi(\xi,\eta,\zeta) d\xi d\eta d\zeta = \sum_i \sum_j \sum_k W_i W_j W_k \phi(\xi,\eta,\zeta) \quad (2.7-3)$$



Gambar 2.4. Posisi *Gauss point*.

Perhitungan integrasi numerik dengan metode *Gauss Quadrature* ini berfungsi pada saat perhitungan tegangan pada elemen dimana untuk kasus elemen isoparametris perhitungan tegangan akan lebih akurat jika dilakukan pada lokasi *Gauss point* dalam elemen, selain itu tingkat keakurasiannya juga dipengaruhi oleh model integrasi atau jumlah aturan *Gauss point* yang akan digunakan pada tiap elemen. Model integrasi atau jumlah *Gauss point* yang biasa digunakan pada analisa adalah *full integration* dengan jumlah *Gauss point*  $m \times m$  dimana  $m$  adalah jumlah node elemen tiap sisi ( $2 \times 2$  untuk dua node pada tiap sisi elemen atau  $3 \times 3$  untuk tiga node pada tiap sisi elemen) sedangkan model integrasi lainnya adalah *reduced integration* dengan jumlah *Gauss point*  $(m-1) \times (m-1)$  ( $1 \times 1$  atau  $2 \times 2$ ).





# **TUGAS AKHIR**

## **BAB III**

### **PENYELESAIAN PERSAMAAN ALJABAR LINEAR SIMULTAN**



### BAB III

## PENYELESAIAN PERSAMAAN ALJABAR LINEAR SIMULTAN

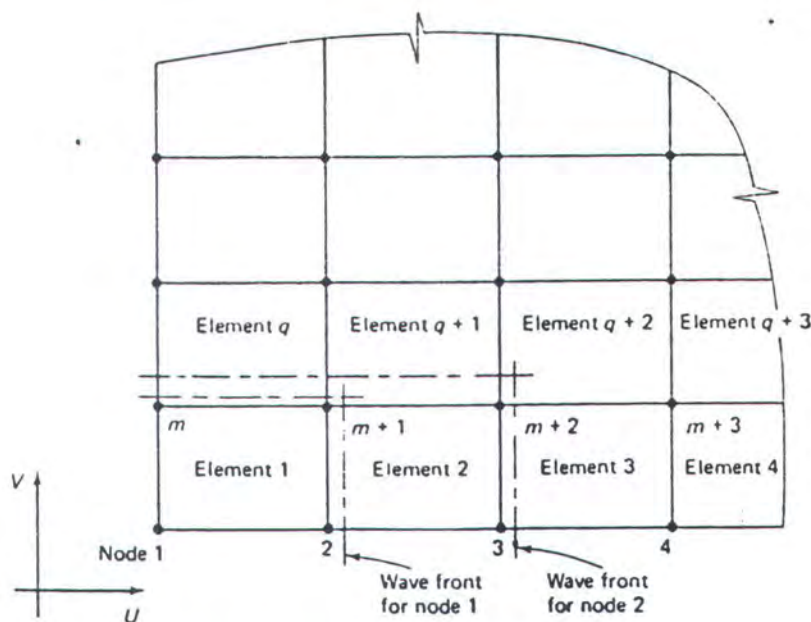
### 3.1. Pendahuluan

Analisa struktur dengan metode elemen hingga tidak akan lepas dari perhitungan komputasi dimana efisiensi perhitungan tersebut dipengaruhi oleh metode penyelesaian persamaan aljabar linear simultan yang terbentuk. Jumlah persamaan aljabar linear simultan yang terbentuk tersebut adalah tergantung dari besarnya jumlah elemen, titik simpul dan derajat kebebasan daripada model struktur yang akan dianalisa.

Penggunaan metode penyelesaian tertentu adalah bertujuan untuk mendapatkan bentuk persamaan yang sedemikian rupa sehingga manipulasi yang harus dilakukan oleh komputer tidak terlalu berkepanjangan, sehingga waktu dan biaya komputasi yang akan dilakukan dapat diperkecil. Sehubungan dengan hal tersebut terdapat beberapa metode yang dapat digunakan yaitu antara lain metode *bandwith*, *wave front*, dan *pointer* dimana konsep pembentukan persamaan dari masing-masing metode adalah tergantung dari penomoran titik simpul ataupun penomoran elemen pada model struktur. Pada permasalahan tugas akhir ini metode yang digunakan adalah metode frontal (*wave front*) dimana hal ini dilakukan karena pertimbangan jumlah elemen, titik simpul dan derajat kebebasan pada model struktur yang akan dianalisa.

### 3.2. Metode Frontal

Teknik frontal merupakan metode yang sangat efektif untuk menyelesaikan masalah struktur dengan banyak elemen maupun derajat kebebasan. Prinsip utama teknik ini adalah *assembly* dan reduksi matrik kekakuan tiap elemen dilakukan dalam waktu yang bersamaan. Reduksi dilakukan dengan cara kondensasi statis dan eliminasi Gauss. Penjelasan teknik tersebut dapat dijelaskan pada analisa *plane stress* pada Gambar 3.1 seperti berikut ini.



Gambar 3.1. Idealisasi penyelesaian frontal.

Tiap elemen mempunyai empat simpul dimana tiap simpul mempunyai dua derajat kebebasan yaitu dispasmen arah  $x$  dan dispasmen arah  $y$ . Selanjutnya proses frontal berjalan sebagai berikut :

- *Assembly* matrik kekakuan elemen 1

Matrik kekakuan elemen berukuran  $8 \times 8$  (4 simpul, 2 d.o.f). Simpul yang tidak berhubungan dengan elemen lain dapat dikondensasi dimana dalam hal ini adalah simpul nomor 1. Dengan kondensasi statis dan eliminasi Gauss dapat



direduksi 2 derajat kebebasan sehingga ukuran matrik kekakuannya sekarang menjadi  $6 \times 6$ .

- *Assembly* matrik kekakuan elemen 2

Karena simpul nomor 1 sudah dinonaktifkan maka ukuran dari *wave front* menjadi  $10 \times 10$  (5 simpul, 2 d.o.f) dan selanjutnya dilakukan eliminasi derajat kebebasan pada simpul yang tidak berhubungan dengan elemen lain dimana dalam hal ini adalah simpul nomor 2 sehingga matrik kekakuan yang sekarang berukuran  $8 \times 8$ .

- Proses berlangsung terus menerus sampai elemen yang terakhir.

Setiap kali penggabungan elemen, saat itu pula dilakukan eliminasi derajat kebebasan pada simpul yang sudah tidak berhubungan dengan elemen lain sehingga dengan demikian matrik kekakuan seluruh struktur secara lengkap tidak akan pernah terbentuk. Ukuran matrik kekakuan sistem tergantung pada *wave front* yang terbentuk dan besarnya *wave front* tergantung dari urutan penomoran elemen yang dilakukan. Jadi dalam menggunakan teknik frontal diperlukan suatu teknik penomoran elemen yang baik sehingga didapatkan ukuran *wave front* yang sekecil mungkin.

### 3.3. Eliminasi Gauss

Prinsip dari penyelesaian sistem persamaan dengan eliminasi Gauss adalah memodifikasi matrik kekakuan menjadi bentuk matrik kekakuan segitiga atas dan kemudian menyelesaikan dengan proses substitusi balik untuk mendapatkan variabel yang dicari.

Persamaan aljabar linear simultan dapat dituliskan dalam bentuk matrik sebagai berikut :

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \dots & K_{1n} \\ K_{21} & K_{22} & K_{23} & \dots & K_{2n} \\ K_{31} & K_{32} & K_{33} & \dots & K_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ K_{n1} & K_{n2} & K_{n3} & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{Bmatrix} = \begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ \dots \\ R_n \end{Bmatrix} \quad (3.3-1)$$

Bentuk matrik tersebut dapat dituliskan dalam bentuk matrik yang dibesarkan (*augmented matrix*) sebagai berikut :

$$\left[ \begin{array}{ccccc|c} K_{11} & K_{12} & K_{13} & \dots & K_{1n} & R_1 \\ K_{21} & K_{22} & K_{23} & \dots & K_{2n} & R_2 \\ K_{31} & K_{32} & K_{33} & \dots & K_{3n} & R_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ K_{n1} & K_{n2} & K_{n3} & \dots & K_{nn} & R_n \end{array} \right] \quad (3.3-2)$$

Selanjutnya langkah-langkah penyelesaian persamaan aljabar linear simultan dengan eliminasi Gauss adalah sebagai berikut :

1. Semua koefisien untuk penyelesaian  $a_1$  pada seluruh persamaan kecuali persamaan baris ke 1 diubah menjadi nol yaitu dengan cara semua unsur elemen baris pertama dikalikan dengan  $-K_{21}/K_{11}$  kemudian dijumlahkan dengan unsur-unsur elemen baris ke 2. Selanjutnya dikalikan semua unsur baris ke 1 lagi dengan  $-K_{31}/K_{11}$  dan jumlahkan dengan unsur-unsur elemen baris ke 3. Proses ini berlanjut hingga baris ke  $n$ . Sehingga hasil dari proses ini akan didapatkan matrik sebagai berikut :



$$\left[ \begin{array}{ccccc|c} K_{11} & K_{12} & K_{12} & \dots & K_{1n} & R_1 \\ 0 & K'_{22} & K'_{23} & \dots & K'_{2n} & R'_2 \\ 0 & K'_{32} & K'_{33} & \dots & K'_{3n} & R'_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & K'_{n2} & K'_{n3} & \dots & K'_{nn} & R'_n \end{array} \right] \quad (3.3-3)$$

2. Semua koefisien untuk penyelesaian  $a_2$  pada semua persamaan kecuali baris ke 1 dan baris ke 2 diubah menjadi nol yaitu dengan cara mengalikan semua unsur elemen baris ke 2 dengan  $-K'_{32}/K'_{22}$  kemudian dijumlahkan pada unsur-unsur elemen baris ke 3. Selanjutnya dikalikan lagi semua unsur elemen baris ke 2 dengan  $-K'_{32}/K'_{22}$  dan dijumlahkan dengan semua unsur-unsur elemen baris ke 4. Proses ini berlanjut hingga baris ke  $n$ . Sehingga hasil dari proses ini mendapatkan bentuk matrik sebagai berikut :

$$\left[ \begin{array}{ccccc|c} K_{11} & K_{12} & K_{12} & \dots & K_{1n} & R_1 \\ 0 & K'_{22} & K'_{23} & \dots & K'_{2n} & R'_2 \\ 0 & 0 & K''_{33} & \dots & K''_{3n} & R''_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & K''_{n3} & \dots & K''_{nn} & R''_n \end{array} \right] \quad (3.3-4)$$

3. Cara yang sama seperti yang tersebut diatas dilakukan untuk penyelesaian selanjutnya hingga baris ke  $n$  sehingga pada proses menghasilkan bentuk matrik yang berupa matrik segitiga atas seperti berikut ini :

$$\left[ \begin{array}{ccccc|c} K_{11} & K_{12} & K_{12} & \dots & K_{1n} & R_1 \\ 0 & K'_{22} & K'_{23} & \dots & K'_{2n} & R'_2 \\ 0 & 0 & K''_{33} & \dots & K''_{3n} & R''_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & K^{(n)}_{nn} & R^{(n)}_n \end{array} \right] \quad (3.3-5)$$

Proses-proses secara keseluruhan untuk mendapatkan matrik segitiga atas dapat dirumuskan secara umum sebagai berikut :

$$K_{ij} = K_{ij} - \frac{K_{ik} K_{kj}}{K_{kk}} \quad (3.3-6)$$

dimana,

$$k = 1, 2, \dots, n-1$$

$$i = k+1, k+2, \dots, n$$

$$j = k, k+1, \dots, n+1$$

$$n = \text{jumlah persamaan}$$

4. Untuk menentukan nilai dari  $a_i$  dapat dihubungkan dengan suatu persamaan berikut ini :

$$a_i = K_{ii} \left( K_{i(n+1)} - \sum_{r=i+1}^n K_{ir} a_r \right) \quad (3.3-7)$$

dimana,

$$i = 1, 2, \dots, n$$

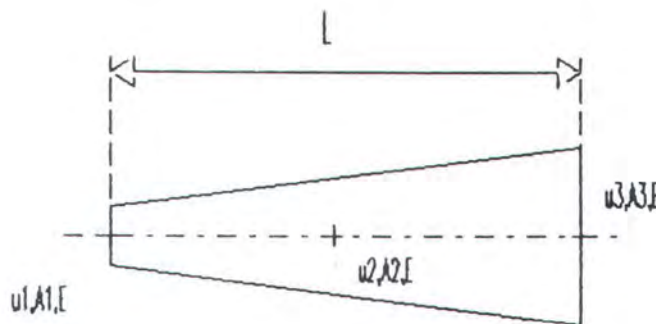
$$r = i+1, i+2, \dots, n$$

$$n = \text{jumlah persamaan}$$



### 3.4. Proses Kondensasi Statis

Pada penyelesaian persamaan aljabar linear simultan dengan teknik frontal, proses kondensasi statis digunakan untuk mereduksi jumlah derajat kebebasan setelah penggabungan satu kekakuan elemen. Penjelasan mengenai hal ini dapat digambarkan pada permasalahan sistem elemen *truss* tiga simpul seperti Gambar 3.2.



Gambar 3.2. Sistem elemen truss.

Simpul nomor 2 pada Gambar 3.2 merupakan simpul yang tidak berhubungan dengan simpul dari elemen yang lain dan karena tidak berhubungan dengan simpul dari elemen yang lain maka simpul nomor 2 ini dapat dieliminasi sehingga akan didapatkan matrik kekakuan yang hanya berhubungan dengan simpul nomor 1 dan simpul nomor 3.

Matrik kekakuan dan displasmen simpul serta vektor beban yang berhubungan dipartisi menjadi bentuk matrik :

$$\begin{bmatrix} K_{aa} & K_{ca} \\ K_{ca} & K_{cc} \end{bmatrix} \begin{Bmatrix} a_a \\ a_c \end{Bmatrix} = \begin{Bmatrix} R_a \\ R_c \end{Bmatrix} \quad (3.4-1)$$

dimana  $a_a$  dan  $a_c$  adalah displasmen yang telah dikondensasi sedangkan matrik  $K_{aa}$ ,  $K_{ac}$ ,  $K_{ca}$ ,  $K_{cc}$  dan vektor beban  $R_a$ ,  $R_c$  berhubungan dengan displasmen  $a_a$  dan  $a_c$ . Dari persamaan 3.4-1 pada baris ke 2, harga  $a_c$  adalah :

$$a_c = K_{cc}^{-1} (R_c - K_{ca} a_a) \quad (3.4-2)$$

harga  $a_c$  pada persamaan di atas disubstitusikan pada persamaan 3.4-1 untuk perhitungan baris ke 1 maka diperoleh suatu persamaan kondensasi :

$$(K_{aa} - K_{ac} K_{cc}^{-1} K_{ca}) a_a = R_a - K_{ac} K_{cc}^{-1} R_c \quad (3.4-3)$$

Karena proses kondensasi derajat kebebasan tersebut di atas adalah dilakukan pada tingkat lokal elemen maka usaha penyusunan matrik kekakuan global struktur dapat dikurangi.





# **TUGAS AKHIR**

## **BAB IV**

### **PERMASALAHAN NON-LINEAR PADA METODE ELEMEN HINGGA**



## BAB IV

### PERMASALAHAN NON-LINEAR PADA METODE ELEMEN HINGGA

#### 4.1. Pendahuluan

Permasalahan elastis-plastis pada mekanika struktur jika ditinjau dengan metode elemen hingga dapat didekati menjadi permasalahan non-linear. Pada permasalahan non-linear, matrik kekakuan dan/atau vektor beban tergantung pada displasmen atau merupakan fungsi dari displasmen [Ref 3]. Secara umum hubungan antara vektor beban, matrik kekakuan dan displasmen dapat ditulis seperti persamaan berikut :

$$[K] \cdot \{D\} = \{R\} \quad (4.1-1)$$

dimana :

$[K]$  = matrik kekakuan

$\{D\}$  = displasmen

$\{R\}$  = vektor beban



Pada analisa linear kedua matrik kekakuan  $[K]$  dan vektor beban  $\{R\}$  adalah tidak tergantung pada displasmen  $\{D\}$ . Sebaliknya pada analisa non-linear matrik kekakuan  $[K]$  dan/atau vektor beban  $\{R\}$  adalah merupakan fungsi dari displasmen  $\{D\}$ .

Ketidaklinearan pada struktur itu sendiri dapat dibedakan menjadi dua, yaitu pertama adalah ketidaklinearan material (*material non-linearity*) dan kedua adalah ketidaklinearan geometri (*geometric non-linearity*). Ketidaklinearan material pada struktur dapat terjadi apabila ada perubahan properti material struktur, hal ini terjadi seperti dalam permasalahan plastisitas dimana perubahan kondisi dari



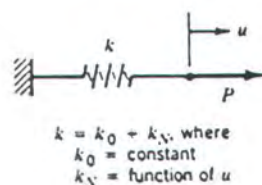
elastis ke plastis menyebabkan perubahan properti material pada struktur. Sedangkan ketidaklinearan geometri pada struktur dapat terjadi karena adanya perubahan bentuk konfigurasi seperti permasalahan *large deflection* pada struktur *slender elastic beam* akibat pembebanan.

#### 4.2. Metode Penyelesaian pada Permasalahan Non-linear

Hubungan antara matrik kekakuan  $[K]$ , vektor beban  $\{R\}$  dan displasmen seperti yang tertulis pada persamaan (4.1-1) jika ditinjau sebagai permasalahan non-linear maka pada permasalahannya dapat dianggap bahwa vektor beban  $\{R\}$  adalah diketahui sedangkan matrik kekakuannya  $[K]$  adalah merupakan fungsi dari displasmen  $\{D\}$  atau matrik kekakuan tersebut dapat dihitung dengan nilai displasmen yang telah diketahui.

Permasalahan di atas dapat dijelaskan seperti pada contoh permasalahan pegas non-linear yang diberi beban  $P$  (Gambar 4.1) dimana kekakuan pegas tersebut terdiri dari kekakuan pegas yang konstan ( $k_0$ ) dan kekakuan pegas yang tergantung pada deformasi ( $k_N$ ). Sedangkan nilai displasmen ( $u$ ) pada struktur pegas tersebut akibat beban  $P$  dihubungkan dengan persamaan sebagai berikut :

$$(k_0 + k_N)u = P \quad \text{dimana } k_N = f(u) \quad (4.2-1)$$



Gambar 4.1. Pegas non-linear.

Dari persamaan tersebut di atas untuk mencari nilai  $u$  dengan diketahuinya nilai beban  $P$  sedangkan nilai  $k_N$  adalah fungsi dari  $u$ , maka penyelesaiannya dilakukan secara iterasi. Metode penyelesaian secara iterasi dari persamaan tersebut mempunyai beberapa metode yang akan dijelaskan pada pembahasan berikut.

#### 4.2.1. Substitusi Langsung

Beban  $P_A$  dikenakan pada struktur pegas dimana pada proses iterasi yang pertama kali diasumsikan bahwa  $k_N = 0$ . Sehingga pendekatan pertama kali untuk nilai displasmen  $u_A$  akibat beban  $P_A$  dapat dihitung dengan  $u_1 = P_A/k_0$ . Hasil  $u_1$  tersebut digunakan untuk menghitung kekakuan yang baru  $k_0 + k_{N1} = k_0 + f(u_1)$ . Selanjutnya untuk menghitung nilai displasmen yang baru  $u_2$  dapat didekati dengan urutan persamaan berikut :

$$u_1 = k_0^{-1} P_A, \quad u_2 = (k_0 + k_{N1})^{-1} P_A, \dots, u_{i+1} = (k_0 + k_{Ni})^{-1} P_A \quad (4.2-2)$$

Dari perhitungan persamaan diatas dapat diinterpretasikan secara grafik seperti pada Gambar 4.2 dimana nilai kekakuan  $(k_0 + k_{Ni})$  dianggap sebagai garis potong pada kurva  $P$  dan  $u$ . Setelah terjadi beberapa kali iterasi maka nilai kekakuan yang merupakan garis potong kurva akan menjadi  $k_0 + k_{NA} \approx P_A/u_A$ , dan hasil penyelesaian displasmen  $u$  yang tepat adalah jika nilai displasmen  $u_A$  mencapai konvergensi atau mendekati dengan nilai  $u$ .

Bentuk alternatif lain dari metode direct substitution adalah jika pada persamaan (4.2-1) nilai  $k_N u$  dipindah ke ruas kanan maka dapat diperoleh urutan persamaan displasmen yang baru sebagai berikut :

$$u_1 = k_0^{-1} P_A, \quad u_2 = k_0^{-1} (P_A - k_{N1} u_1), \dots, u_{i+1} = k_0^{-1} (P_A - k_{Ni} u_i) \quad (4.2-3)$$

Pada kedua persamaan (4.2-2) dan (4.2-3) terdapat perbedaan untuk hasil  $u_1, u_2, \dots$ , dst., tetapi saat konvergensi maka pada kedua persamaan akan menghasilkan  $u_\infty \approx u_A$ . Persamaan 4.2-3 dapat diinterpretasikan secara grafik seperti pada

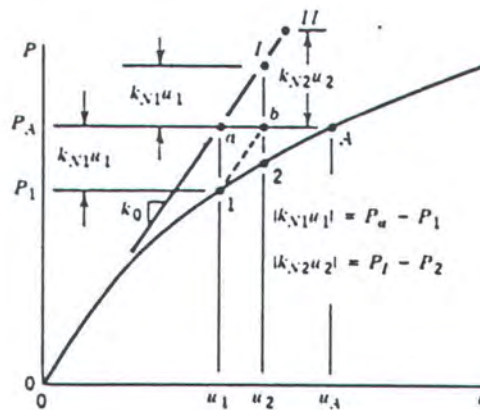
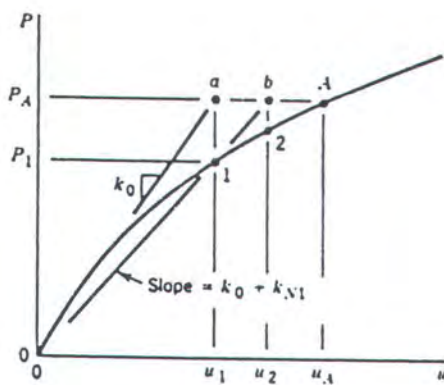


Gambar 4.3. Pada Gambar tersebut dapat diperoleh hubungan dari beban efektif yang bekerja pada proses iterasi kedua dan ketiga yaitu seperti persamaan berikut :

$$P_A - k_{N1}u_1 = P_A + (k_0u_1 - [k_0 + k_{N1}]u_1) = P_A + (P_a - P_1) \quad (4.2-4)$$

$$P_A - k_{N2}u_2 = P_A + (k_0u_2 - [k_0 + k_{N2}]u_2) = P_A + (P_I - P_2) \quad (4.2-5)$$

Sebagai catatan, diketahui bahwa nilai  $P_a - P_I = P_I - P_A$  dan  $P_I - P_2 = P_{II} - P_A$  hal ini dapat terlihat pada Gambar 4.3. Penyelesaian yang konvergen untuk mendapatkan displasmen  $u \approx u_A$  adalah jika penyelesaian  $P_a - P_I$ ,  $P_a - P_I, \dots$ , dst. juga konvergen.



Gambar 4.2. Interpretasi grafik dari pers. 4.2-2.      Gambar 4.3. Interpretasi grafik dari pers. 4.2-3.

#### 4.2.2. Metode Newton Raphson (N-R)

Jika beban  $P_A$  diberikan pada struktur pegas maka persamaan 4.2-1 dalam hal ini persamaannya dapat berubah menjadi :

$$(k_0 + k_{NA})u_A = P_A \quad \text{dimana} \quad k_{NA} = f(u_A) \quad (4.2-6)$$

Selanjutnya beban akan dinaikkan menjadi  $P_B$  dan nilai displasmen yang baru  $u_B$  juga akan dicari, dengan ekspansi deret Taylor pada penyelesaian fungsi  $P = f(u)$  untuk displasmen  $u_A$  diperoleh suatu persamaan :

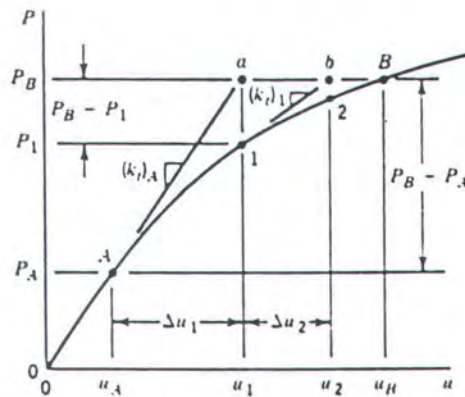
$$f(u_A + \Delta u_1) = f(u_A) + \left( \frac{dP}{du} \right)_A \Delta u_1 \quad (4.2-7)$$

dimana

$$\left( \frac{dP}{du} \right)_A = \frac{d}{du} (k_0 u + k_N u) = k_0 + \frac{d}{du} (k_N u) = k_t \quad (4.2-8)$$

$k_t$  pada persamaan diatas disebut sebagai *tangent stiffness* dan merupakan garis singgung dari kurva grafik pada Gambar 4.4, untuk perhitungan  $\Delta u_1$  terlihat pada Gambar 4.4 bahwa  $f(u_A + \Delta u_1) = P_B$  dan  $f(u_A) = P_A$  sehingga persamaan 4.2-8 menjadi :

$$P_B = P_A + (k_t)_A \Delta u_1 \text{ atau } (k_t)_A \Delta u_1 = P_B - P_A \quad (4.2-9)$$



Gambar 4.4. Interpretasi grafik pers. N-R.

dimana  $P_B - P_A$  merupakan ketidakseimbangan gaya dan hasil perhitungan  $\Delta u_1$  digunakan untuk menghitung nilai displasmen yang baru  $u_2 = u_A + \Delta u_1$ . Untuk proses iterasi selanjutnya akan didapatkan nilai  $(k_t)_1$  yang baru, didapat dari persamaan 4.2-8 dengan nilai  $u = u_1$  dan juga nilai ketidakseimbangan gaya yang baru  $P_B - P_1$ , dimana  $P_1$  didapat dari persamaan 4.2-1 dengan nilai  $u = u_1$ . Sedangkan displasmen yang baru selanjutnya dapat dihitung melalui  $u_2 = u_A + \Delta u_1$ ,



dimana  $\Delta u_1$  diperoleh dari penyelesaian persamaan  $(k_t)_1 \Delta u_2 = P_B - P_1$ . Selanjutnya penyelesaian nilai displasmen yang sebenarnya adalah ketika nilai ketidakseimbangan gaya mencapai konvergensi atau mendekati nol.

#### 4.2.3. Metode Incremental

Metode penyelesaian ini dapat dijelaskan dengan metode Euler untuk penyelesaian persamaan diferensial orde pertama. Hal ini dapat dijelaskan jika pada persamaan 4.2-1,  $P$  adalah merupakan fungsi dari  $u$  atau  $P = f(u)$  dan didefinisikan  $k_t = dP/du$  selain itu apabila proses incremental pembebanan  $\Delta P$  dimulai dari  $P = 0$  dan  $u = 0$ , maka penyelesaiannya dapat dihitung melalui persamaan sebagai berikut :

$$u_1 = 0 + (k_t)_0^{-1} \Delta P_1 \quad \text{dimana } (k_t)_0 = k_t \text{ pada } u = 0 \quad (4.2-10)$$

$$u_2 = u_1 + (k_t)_1^{-1} \Delta P_2 \quad \text{dimana } (k_t)_1 = k_t \text{ pada } u = u_1 \quad (4.2-11)$$

$$u_3 = u_2 + (k_t)_2^{-1} \Delta P_3 \quad \text{dimana } (k_t)_2 = k_t \text{ pada } u = u_2 \quad (4.2-12)$$

dan secara umum persamaan-persamaan diatas dapat ditulis sebagai berikut :

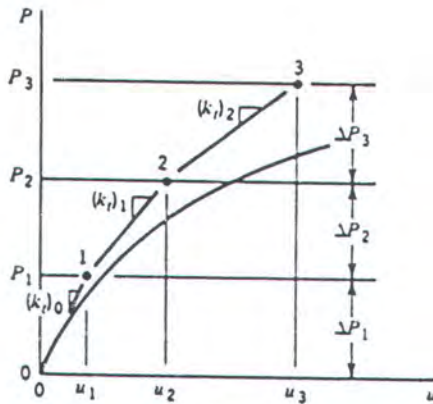
$$u_{i+1} = u_i + (k_t)_i^{-1} \Delta P_{i+1} \quad (4.2-13)$$

Persamaan tersebut dapat diinterpretasikan secara grafik seperti pada Gambar 4.5 dimana pada Gambar terlihat kelemahan dari metode incremental yaitu hasil penyelesaiannya tidak mendekati dengan hasil yang sebenarnya pada tiap proses penambahan beban. Keadaan ini dapat dikurangi dengan menambahkan faktor korektif beban sehingga persamaan 4.2-13 berubah menjadi :

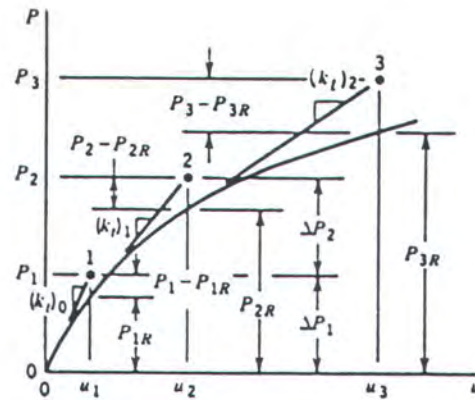
$$u_{i+1} = u_i + (k_t)_i^{-1} \{ \Delta P_{i+1} + (P_i - P_{iR}) \} \quad (4.2-14)$$

dimana  $P_i$  adalah total beban luar yang bekerja pada saat proses incremental  $i$  atau  $P_i = \sum \Delta P_i$ , sedangkan  $P_{iR}$  adalah beban yang ditahan oleh pegas akibat beban luar

tersebut atau  $P_{iR} = (k_0 + k_{Ni})u_i$ . Interpretasi secara grafik dari persamaan 4.2-14 adalah seperti pada Gambar 4.6.



Gambar 4.5. Interpretasi grafik pers. 4.2-13.



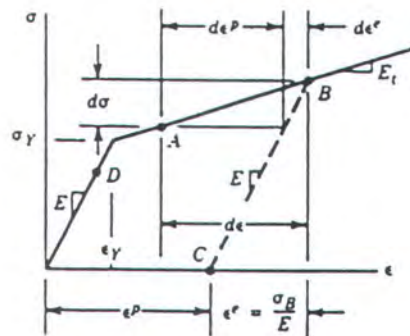
Gambar 4.6. Interpretasi grafik pers. 4.2-14.

Sebagai catatan bahwa untuk permasalahan struktur yang derajat kebebasannya adalah multidimensional maka dalam hal ini  $k$  atau kekakuan struktur adalah merupakan bentuk matrik. Sehingga pada persamaan 4.2-1, kekakuan struktur merupakan matrik dari  $[K_0 + K_N]$  dan selain itu beban  $P$  dalam hal ini berubah menjadi  $\{R\}$  dan displasmen  $u$  berubah menjadi  $\{D\}$ .

### 4.3. Permasalahan Elastis-Plastis

Material pada struktur dianggap non-linear apabila hubungan antara tegangan ( $\sigma$ ) dan regangan ( $\epsilon$ ) dipengaruhi oleh suatu matrik yang tidak konstan, yaitu matrik yang tergantung pada regangan (*strain-dependent matrix*). Hal ini terdapat pada permasalahan struktur dimana properti material struktur tersebut mengalami perubahan pada saat terjadi transisi dari kondisi elastis menjadi plastis sehingga dalam hal ini aliran plastis (*plastic flow*) mulai berpengaruh pada material.





Gambar 4.7. Hubungan tegangan-regangan.

Perubahan kondisi elastis menjadi plastis pada struktur diawali dengan terjadinya proses peluluhan (*yielding*) dari tegangan material. Pada Gambar 4.7 terlihat bahwa setelah proses *yielding* terjadi maka penambahan regangan  $d\varepsilon$  yang terjadi merupakan komposisi regangan dari kondisi elastis  $d\varepsilon^e$  dan dari kondisi plastis  $d\varepsilon^p$  atau  $d\varepsilon = d\varepsilon^e + d\varepsilon^p$ . Sedangkan penambahan tegangan  $d\sigma$  dapat dihubungkan dengan persamaan-persamaan berikut :

$$d\sigma = E(d\varepsilon - d\varepsilon^p) \quad d\sigma = E_t d\varepsilon \quad \text{dan} \quad d\sigma = H d\varepsilon^p \quad (4.3-1)$$

dimana  $H$  adalah parameter *hardening* dan  $E_t$  adalah *tangent modulus*. Jika pada persamaan (4.3-1), persamaan pertama dan ketiga disubstitusikan ke dalam persamaan kedua maka akan diperoleh suatu persamaan berikut :

$$H = \frac{E}{1 - (E_t / E)} \quad \text{atau} \quad E_t = E \left( 1 - \frac{E}{E + H} \right) \quad (4.3-2)$$

dan jika dalam hal ini  $E$  mempunyai nilai tertentu tetapi  $E_t = 0$  sehingga  $H = 0$  maka kondisi material dapat disebut *elastic-perfectly plastic*. Sedangkan kriteria *yield* pada permasalahan ini adalah ketika tegangan  $|\sigma|$  telah mencapai tegangan *yield* material struktur  $\sigma_Y$ .

Aplikasi permasalahan elastis-plastis tersebut diatas dapat dijelaskan pada contoh berikut, dimana pada Gambar 4.8 terlihat suatu struktur bar yang diberikan beban aksial  $P$  dan properti material struktur tersebut adalah seperti yang digambarkan pada Gambar 4.7. Jika struktur tersebut dimodelkan memiliki dua derajat kebebasan untuk tiap elemen maka matrik kekakuan singgung (*tangent stiffness*) dari elemen dapat ditulis :

$$[k_t] = \frac{AE_{ep}}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (4.3-3)$$

dimana  $E_{ep} = E$  jika kriteria *yield* dari struktur belum tercapai atau jika struktur dalam keadaan belum diberi beban, dan  $E_{ep} = E_t$  jika kriteria *yield* sudah terlampaui dan aliran plastis sudah mempengaruhi material struktur. Pada saat penyelesaian permasalahan di atas, dapat juga terjadi proses transisi dalam material dari kondisi elastis menjadi plastis sehingga penyelesaian  $E_{ep}$  dalam hal ini adalah merupakan kombinasi dari  $E$  dan  $E_t$  yang disesuaikan dengan nilai fraksi  $m$ . Hal ini dapat dijelaskan dengan Gambar 4.7 dimana proses tersebut terjadi di antara  $\epsilon_D$  dan  $\epsilon_A$  sehingga penyelesaian  $E_{ep}$  menjadi :

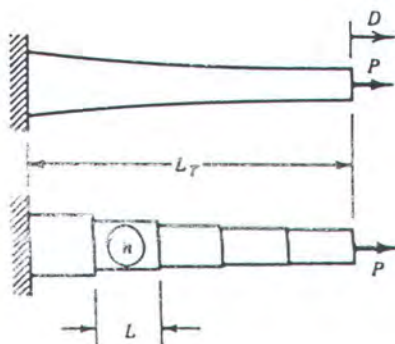
$$E_{ep} = mE + (1 - m)E_t \quad \text{dimana } m = \frac{\epsilon_Y - \epsilon_D}{\epsilon_A - \epsilon_D} \quad (4.3-4)$$

Sedangkan prosedur penyelesaian masalah untuk mendapatkan hubungan antara pembebanan dengan displasmen dan penentuan tegangan elemen pada contoh di atas akibat proses incremental pembebanan  $\Delta P$ , dapat dijelaskan seperti pada algoritma dibawah ini yaitu sebagai berikut :

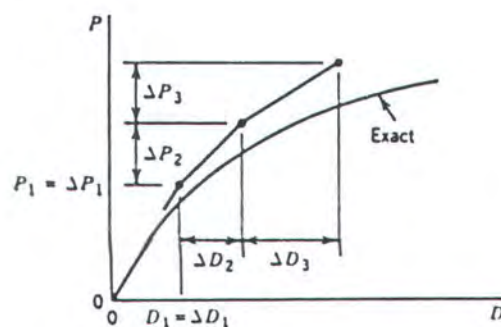
1. Pada proses perhitungan yang pertama kali ( $i = 1$ ), diasumsikan  $E_{ep} = E$  untuk tiap elemen dan struktur diberikan beban incremental untuk yang pertama kalinya  $\Delta P_1 = \Delta R_1$ .



2. Periksa hubungan tegangan-regangan untuk mengevaluasi  $E_{ep}$  pada tiap elemen dan dengan persamaan 4.3-3 dihitung  $[k_t]_n$  untuk tiap elemen  $n$  dimana selanjutnya dari dapat diperoleh matrik kekakuan singgung (*tangent stiffness*) struktur  $[K_t]_{i-1}$ . Setelah itu dilakukan perhitungan  $[K_t]_{i-1} \{\Delta D\}_i = \{R\}_i$  dan dari hasil perhitungan  $\{\Delta D\}_i$  dapat diperoleh  $\Delta \epsilon_i$ .
4. Tanpa perubahan beban  $\{R\}_i$  diulangi langkah di atas hingga penyelesaian mencapai konvergen.
4. Setelah hasil penyelesaian di atas mencapai konvergen maka selanjutnya dilakukan perhitungan nilai-nilai yang baru untuk  $\{D\}_i = \{D\}_{i-1} + \{\Delta D\}_i$ , dan untuk tiap elemen  $\epsilon_i = \epsilon_{i-1} + \Delta \epsilon_i$  dan  $\sigma_i = \sigma_{i-1} + \Delta \sigma_i$  dimana  $\Delta \sigma_i = (E_{ep})_i \Delta \epsilon_i$ . Untuk proses perhitungan  $i = 1$  dan perhitungan dimulai pada konfigurasi struktur yang tanpa beban, maka nilai initial  $i = i - 1$  untuk displasmen, tegangan dan regangan adalah nol.
5. Memberikan beban incremental selanjutnya dan dilakukan proses konvergensi dengan iterasi.
6. Proses incremental berhenti setelah  $\Sigma(\Delta R)_i$  mencapai total beban yang dimaksud.



Gambar 4.8. Struktur bar dengan model finite elemennya

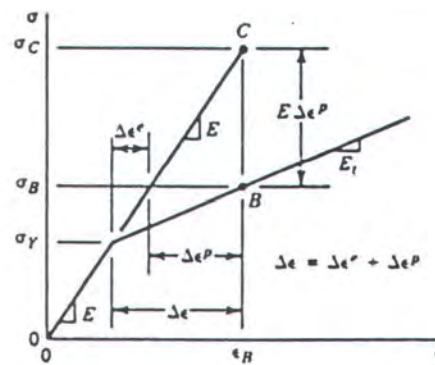


Gambar 4.9. Proses penyelesaian metode tangent-stiffness

Gambar 4.9 adalah merupakan proses penyelesaian dari permasalahan di atas dimana perhitungan konvergensi berdasarkan pada ketidakseimbangan gaya (*load imbalance*) yang terjadi.

#### 4.4. Hubungan Tegangan-Regangan pada Plastisitas

Teori plastisitas dapat terdiri dari tiga bagian, yaitu kriteria *yield*, *flow rule* dan *hardening rule*. Secara umum dalam teorinya, bentuk-bentuk persamaan yang ada adalah merupakan pendekatan dari hasil eksperimen tentang plastisitas.



Gambar 4.10. Hubungan tegangan regangan pada kondisi *yield*

Pada kriteria *yield* didefinisikan suatu fungsi *yield* yang merupakan fungsi dari tegangan  $\{\sigma\}$  dan faktor kuantitas  $\{\alpha\}$  dan  $W_p$  dimana kedua faktor tersebut berhubungan dengan permasalahan *hardening rule*. Kondisi *yield* terjadi jika fungsi *yield* sama dengan nol atau dapat dituliskan sebagai berikut :

$$F = f(\sigma, \alpha, W_p) - \sigma_Y = 0 \quad (4.4-1)$$

dimana  $\{\alpha\}$  dan  $W_p$  didefinisikan pada persamaan 4.4-4. Pada dasarnya evaluasi dari fungsi  $F$  dengan diketahuinya nilai  $\{\sigma\}$ ,  $\{\alpha\}$  dan  $W_p$  hanya akan memberikan



hasil  $F < 0$  dan  $F = 0$  dimana masing-masing menunjukkan kondisi bahwa material dalam keadaan elastis dan mengalami *yield*. Sedangkan penyelesaian fungsi *yield* di atas yang memberikan hasil  $F > 0$ , secara kenyataan adalah tidak mungkin karena hal ini memberikan indikasi bahwa hubungan tegangan yang terjadi tidak memenuhi hukum yang ada (seperti terlihat pada Gambar 4.10 bahwa  $\sigma_C$  tidak akan mungkin terjadi).

*Flow rule* pada teori plastisitas digunakan untuk mendefinisikan nilai potensial plastis  $Q$  yang merupakan satuan dan fungsi dari tegangan dan diasumsikan bahwa untuk permasalahan plastisitas pada mekanika struktur metal adalah  $Q \approx F = f(\sigma, \alpha, W_p)$ . Dengan  $d\lambda$  sebagai faktor pengali plastis (*plastic multiplier*) maka akan didapatkan persamaan penambahan regangan plastis  $d\epsilon^p$  sebagai berikut :

$$\{d\epsilon^p\} = \left\{ \frac{\partial Q}{\partial \sigma} \right\} d\lambda \quad (4.4-2)$$

Sedangkan *hardening rule* dalam hal ini digunakan untuk mendefinisikan faktor kuatitas  $\{\alpha\}$  dan  $W_p$  sebagai berikut :

$$\{\alpha\} = \int C \{d\epsilon^p\} \quad \text{dan} \quad W_p = \int (\sigma)^T \{d\epsilon^p\} \quad (4.4-3)$$

dimana  $C$  diasumsikan sebagai suatu konstanta material dan  $W_p$  dapat diidentifikasikan sebagai *plastic work* per unit volume.

Hubungan tegangan-regangan pada plastisitas adalah analog dengan hubungan tegangan-regangan pada elastisitas yaitu  $\{\sigma\} = [E]\{\epsilon\}$ . Hal ini dapat dijelaskan pada penurunan persamaan-persamaan sebagai berikut, pertama kali dari diferensiasi persamaan 4.4-1 dapat diperoleh persamaan :

$$dF = 0 = \left\{ \frac{\partial F}{\partial \sigma} \right\}^T \{d\sigma\} + \left\{ \frac{\partial F}{\partial \alpha} \right\}^T \{d\alpha\} + \frac{\partial F}{\partial W_p} dW_p \quad (4.4-4)$$

Sedangkan dari penurunan persamaan 4.4-3 dan analog dengan persamaan 4.3-1 diperoleh persamaan-persamaan sebagai berikut :

$$\{d\alpha\} = C\{d\varepsilon^p\}, \quad dW_p = \{\sigma\}^T \{d\varepsilon^p\} \quad (4.4-5)$$

$$\text{dan} \quad \{d\sigma\} = [E]\{d\varepsilon^e\} = [E](\{d\varepsilon\} - \{d\varepsilon^p\}) \quad (4.4-6)$$

Jika dengan persamaan 4.4-2 dan persamaan-persamaan di atas disubstitusikan pada persamaan 4.4-4 maka akan didapatkan persamaan *plastic multiplier*  $d\lambda$ , sebagai berikut :

$$d\lambda = \{C_\lambda\}^T \{d\varepsilon\} \quad (4.4-7)$$

dimana

$$\{C_\lambda\}^T = \frac{\left\{ \frac{\partial F}{\partial \sigma} \right\}^T [E]}{\left\{ \frac{\partial F}{\partial \sigma} \right\}^T [E] \left\{ \frac{\partial Q}{\partial \sigma} \right\} - C \left\{ \frac{\partial F}{\partial \alpha} \right\}^T \left\{ \frac{\partial Q}{\partial \sigma} \right\} - \frac{\partial F}{\partial W_p} \{\sigma\}^T \left\{ \frac{\partial Q}{\partial \sigma} \right\}} \quad (4.4-8)$$

dan jika persamaan 4.4-2 disubstitusikan pada persamaan 4.4-6 maka akan diperoleh persamaan sebagai berikut :

$$\{d\sigma\} = [E] \left( \{d\varepsilon\} - \left\{ \frac{\partial Q}{\partial \sigma} \right\} d\lambda \right) \quad \text{atau} \quad \{d\sigma\} = [E_{ep}]\{d\varepsilon\} \quad (4.4-9)$$

dimana



$$[E_{ep}] = [E] - [E] \left\{ \frac{\partial Q}{\partial \sigma} \right\} \{C_\lambda\}^T \quad (4.4-10)$$

Persamaan 4.4-10 di atas merupakan generalisasi dari persamaan *tangent modulus*  $E_t$ , seperti pada persamaan 4.3-2 yang terdahulu dan persamaan tersebut dapat digunakan pada generalisasi matrik *tangent stiffness*  $[k_t]$  menjadi :

$$[k_t] = \int_V [B]^T [E_{ep}] [B] dV \quad (4.4-11)$$

dimana pada persamaan atas nilai  $[E_{ep}]$  adalah seperti pada persamaan 4.4-10 jika  $F = 0$  dan  $dF = 0$ , tetapi jika  $F < 0$  dan  $dF < 0$  maka nilai  $[E_{ep}]$  dalam hal ini adalah merupakan nilai modulus elastisitas  $[E]$ .

#### 4.5. Prosedur Analisa Elastis-Plastis

Pada pembahasan sebelumnya telah dibahas prosedur analisa elastis-plastis untuk permasalahan struktur non-multi dimensional (satu dimensi). Sedangkan pada pembahasan berikut ini merupakan generalisasi dari permasalahan sebelumnya dimana kondisi multi-aksial mempengaruhi hubungan tegangan-regangan dari pada struktur. Salah satu metode yang digunakan untuk prosedur analisa elastis-plastis ini adalah metode *tangent stiffness* dan algoritma penyelesaian dari metode tersebut adalah seperti berikut :

1. Total beban  $\{R\}$  yang akan dikenakan pada struktur, diberikan dalam bentuk incremental  $\{\Delta R_i\}$  sedemikian hingga  $\{R\} = \Sigma \{\Delta R_i\}$ . Pada saat struktur pertama kali diberi beban  $\{\Delta R_1\}$  dari keadaan tanpa pembebanan sebelumnya, maka dapat ditetapkan bahwa  $\{\epsilon\} = \{\sigma\} = \{\alpha\} = W_p = 0$  dan  $[E_{ep}] = [E]$  untuk tiap elemen.
2. Gunakan hasil dari  $\{\sigma\}_{i-1}$ ,  $\{\alpha\}_{i-1}$  dan  $W_{pi-1}$  untuk mengevaluasi  $[E_{ep}]_{i-1}$  dimana  $[E_{ep}]_{i-1} = [E]$  jika kondisi elemen belum mencapai *yield* ( $F < 0$ ) atau dalam keadaan tanpa beban ( $dF < 0$ ). Selanjutnya hitung matrik kekakuan  $[k_t]$

untuk tiap elemen  $n$  yang kemudian digunakan untuk menyusun matrik *tangent stiffness* untuk keseluruhan struktur,  $[K_L]_{i-1} = \Sigma [k_L]_n$ . Hitung displasmen struktur  $\{\Delta D\}_i$  dan penambahan regangan yang terjadi  $\{\Delta \epsilon\}_i$  dengan persamaan berikut :

$$[K_L]_{i-1} \{\Delta D\}_i = \{R\}_{i-1} \quad \text{dan} \quad \{\Delta \epsilon\}_i = [B] \{\Delta D\}_i \quad (4.5-1)$$

Jika kondisi elemen mencapai plastis maka perhitungan incremental yang dilakukan adalah seperti persamaan berikut :

$$\text{dari persamaan 4.4-7} \quad \Delta \lambda_i = \int \{C_\lambda\}^T \{d\epsilon\} \approx \{C_\lambda\}_{i-1}^T \{\Delta \epsilon\}_i \quad (4.5-2)$$

$$\text{dari persamaan 4.4-2} \quad \{\Delta \epsilon^p\}_i = \int \left\{ \frac{\partial Q}{\partial \sigma} \right\} d\lambda \approx \left\{ \frac{\partial Q}{\partial \sigma} \right\}_{i-1} \Delta \lambda_i \quad (4.5-3)$$

$$\text{dari persamaan 4.4-6} \quad \{\Delta \sigma\}_i = [E] (\{\Delta \epsilon\}_i - \{\Delta \epsilon^p\}_i) \quad (4.5-4)$$

$$\text{dari persamaan 4.4-3} \quad \{\Delta \alpha\}_i = \int C \{d\epsilon^p\} \approx C \{\Delta \epsilon^p\}_i \quad (4.5-5)$$

$$\text{dari persamaan 4.4-3} \quad \Delta W_{pi} = \int \{\sigma\}^T \{d\epsilon^p\} = \{\sigma\}_i^T \{\Delta \epsilon^p\}_i \quad (4.5-6)$$

4. Jika pada saat penyelesaian kondisi elemen mengalami transisi dari kondisi elastis menjadi plastis maka evaluasi  $[E_{ep}]$  dilakukan dengan persamaan sebagai berikut :

$$\text{dari persamaan 4.4-10} \quad [E_{ep}] = [E] - (1-m)[E] \left\{ \frac{\partial Q}{\partial \sigma} \right\}_{i-1} \{C_\lambda\}_{i-1}^T \quad (4.5-7)$$

dimana  $m$  adalah nilai fraksi yang dihitung dengan menggunakan kondisi tegangan pada saat itu  $Y_i$



$$m = \frac{\sigma_r - Y_{i-1}}{Y_i - Y_{i-1}} \quad (4.5-8)$$

Sebelum proses incremental beban selanjutnya dilakukan maka perhitungan pada langkah ke-2 sampai ke-3 tersebut di atas terus dilakukan hingga penyelesaiannya mencapai konvergen.

4. Setelah penyelesaian mencapai konvergen maka selanjutnya dilakukan perhitungan untuk nilai-nilai yang baru seperti :

$$\begin{aligned} \{D\}_i &= \{D\}_{i-1} + \{\Delta D\}_i & \{\sigma\}_i &= \{\sigma\}_{i-1} + \{\Delta \sigma\}_i \\ \{\alpha\}_i &= \{\alpha\}_{i-1} + \{\Delta \alpha\}_i & \{W_p\}_i &= \{W_p\}_{i-1} + \{\Delta W_p\}_i \end{aligned} \quad (4.5-9)$$

5. Proses incremental pembebanan selanjutnya dapat dilakukan dengan memberikan faktor korektif beban seperti yang telah dijelaskan sebelumnya, sehingga persamaan incremental beban menjadi

$$\{\Delta R\}_{i+1} + \{\Delta R_c\}_i \quad \text{dimana} \quad \{\Delta R_c\}_i = \{R\}_i - \sum \int_V [B]^T \{\sigma\}_i dV \quad (4.5-10)$$

dan  $\{R\}_i$  adalah total beban eksternal yang bekerja pada saat itu.

6. Keseluruhan proses incremental berhenti jika  $\Sigma \{\Delta R\}_i$  mencapai total beban  $\{R\}$ .





# **TUGAS AKHIR**

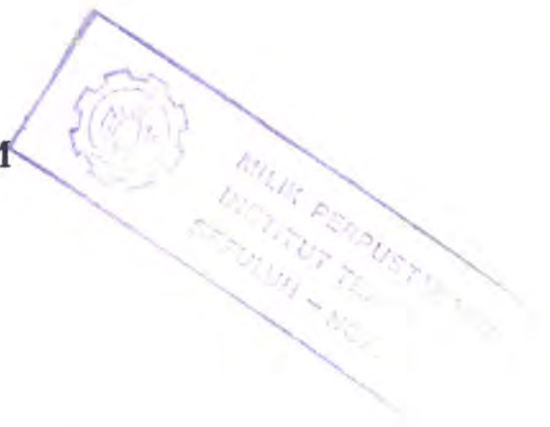
**BAB V**

**APLIKASI PROGRAM**



## BAB V

### APLIKASI PROGRAM



#### 5.1. Pendahuluan

Penulisan program untuk menganalisa permasalahan non-linear yang ada dalam tugas akhir ini merupakan hasil modifikasi dari referensi program tertulis yang disusun oleh *J. A. FIGUEIRAS* dan *D. R. J. OWEN* [Ref. 2]. Pada tugas akhir program disusun dalam bahasa FORTRAN dimana penulisan dan penyusunannya didukung oleh fasilitas program MS FORTRAN versi 5.10 dan hasil dari penyusunan program tersebut merupakan suatu program kerja yang langsung dapat dieksekusi (*\_exe-file*).

Penulisan program untuk analisa permasalahan non-linear tersebut di atas adalah terdiri dari satu program utama dan tiga puluh dua program subroutine dan sebagai program tambahan untuk analisa grafis dari model elemen sebelum maupun sesudah deformasi maka disusun satu program kerja tersendiri dimana input programnya merupakan hasil output dari program utama sebelumnya.

#### 5.2. Program Utama

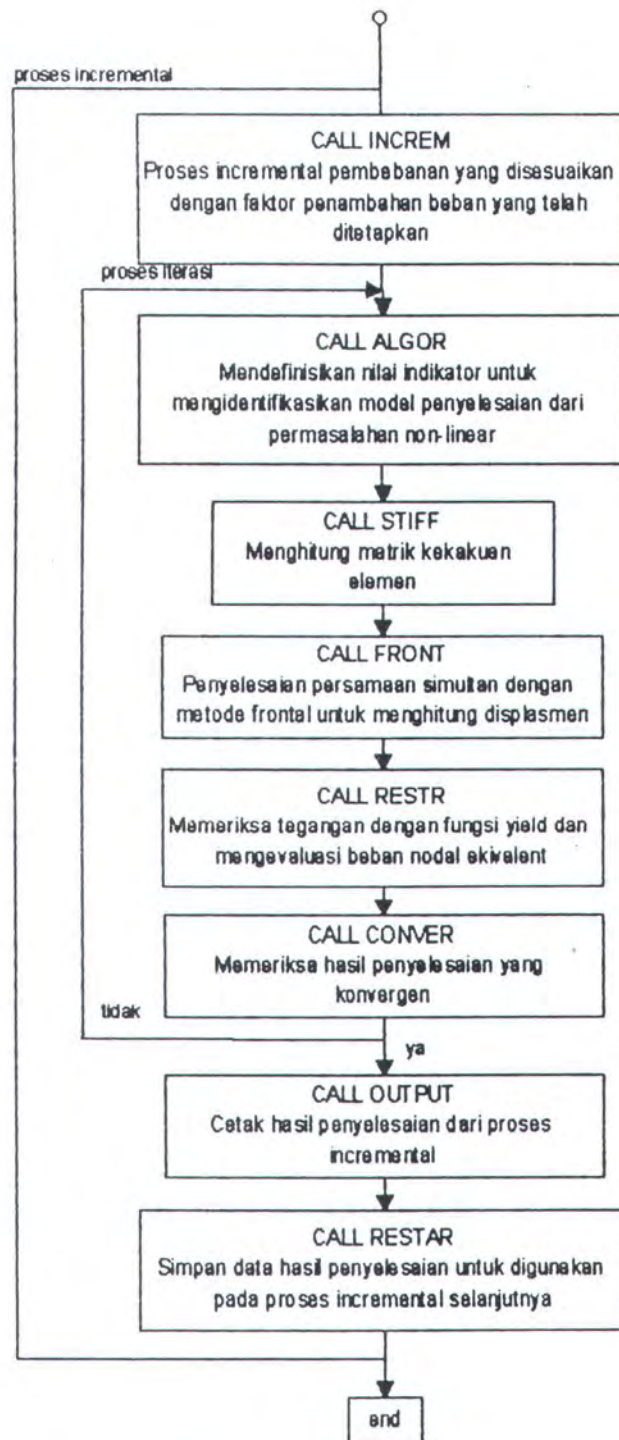
Program utama ini berfungsi sebagai program yang mengontrol dan memanggil program-program subroutine yang dibutuhkan pada saat proses program dilakukan. Selain itu program utama juga berfungsi sebagai pengontrol proses incremental dan proses iterasi yang terjadi pada keseluruhan proses program.

Struktur dari program utama ini dapat dijelaskan secara skematik oleh blok diagram sebagai berikut :



Gambar 5.1. Flow chart program utama.





Gambar 5.2. Lanjutan *flow chart* program utama.

### 5.3. Program Subroutine

Keseluruhan program subroutine yang digunakan adalah berjumlah tiga puluh dua dimana pemanggilannya selain melalui program utama, juga ada beberapa

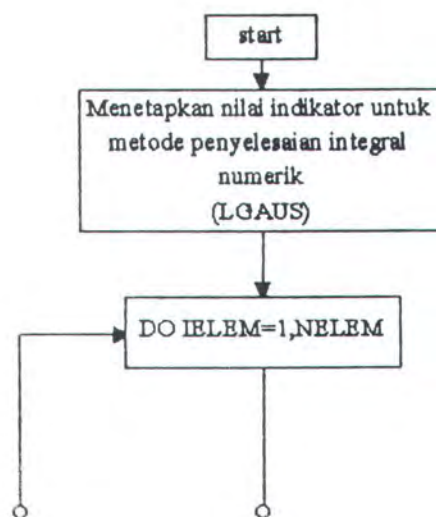
program subroutine yang dipanggil melalui program subroutine lain. Penjelasan secara garis besar mengenai program-program subroutine yang digunakan adalah sebagai berikut :

1. Subroutine ALGOR

Program subroutine ini berfungsi menetapkan nilai indikator untuk mengidentifikasi tipe penyelesaian dari proses non-linear dimana nilai indikator tersebut dapat mengidentifikasi metode perhitungan dari matrik kekakuan pada tiap proses incremental maupun proses iterasi.

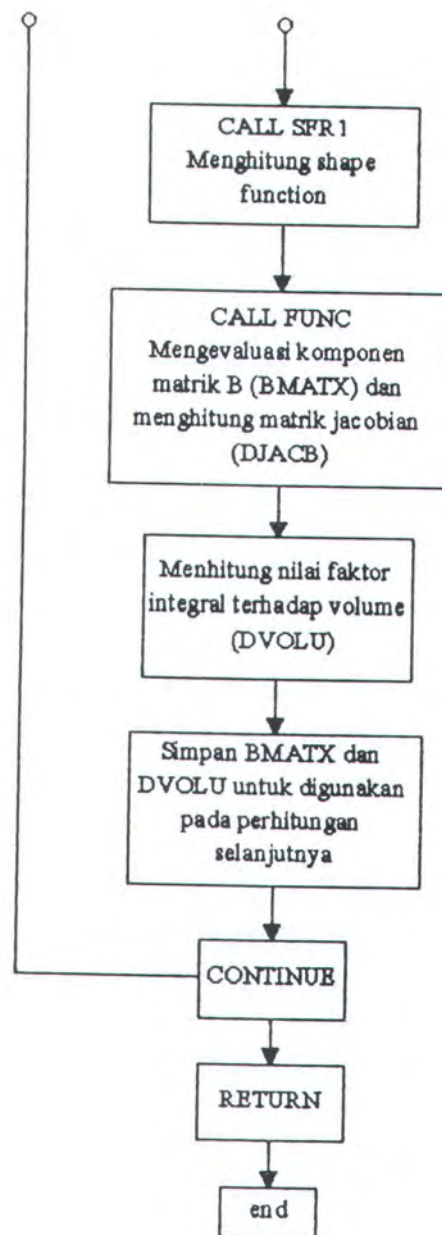
2. Subroutine BGMAT

Program subroutine ini berfungsi menghitung matrik B dan mendefinisikan metode penyelesaian integral numerik dimana hasil data yang diperoleh akan disimpan untuk digunakan pada proses perhitungan selanjutnya. Secara garis besar proses yang terjadi pada program subroutine adalah sebagai berikut :



Gambar 5.3. Flow chart Subroutine BGMAT.





Gambar 5.4. Lanjutan *flow chart* Subroutine BGMAT

### 3. Subroutine CHECK1

Program subroutine ini berfungsi sebagai pengontrol kesalahan yang mungkin terjadi dari parameter-parameter perhitungan akibat kesalahan data input program. Kesalahan yang terdeteksi oleh subroutine ini akan ditunjukkan dengan

jenis nomor kesalahan dan keterangan maksud dari nomor kesalahan tersebut adalah seperti berikut ini :

- 1 Input dari keseluruhan nodal point (NPOIN) adalah nol atau kurang dari nol.
- 2 Total dari nodal point pada keseluruhan elemen adalah kurang dari NPOIN ( $NELEM * NNODE < NPOIN$ ).
- 3 Total restrained nodal point adalah melebihi NPOIN.
- 4 Total jumlah node pada elemen adalah tidak sama dengan 8 atau 9.
- 5 Total jumlah d.o.f tidak sama dengan 5 dan jumlah layer untuk tiap elemen adalah lebih dari 10.
- 6 Total jumlah jenis material adalah kurang atau sama dengan nol atau lebih dari jumlah maksimum yang telah ditentukan (NMATS).
- 7 Jumlah Gauss point per elemen (bending, membran) tidak sama dengan 2 atau 3.
- 8 Jumlah Gauss point per elemen (shear) tidak sama dengan 2 atau 3.

#### 4. Subroutine CHECK2

Setelah proses dari subroutine CHECK1 terlewati maka Program subroutine CHECK2 akan memeriksa keabsahan dari data geometri elemen, kondisi batas dan properti material. Kesalahan yang terdeteksi akan ditunjukkan dengan nomor-nomor kesalahan sebagai berikut :

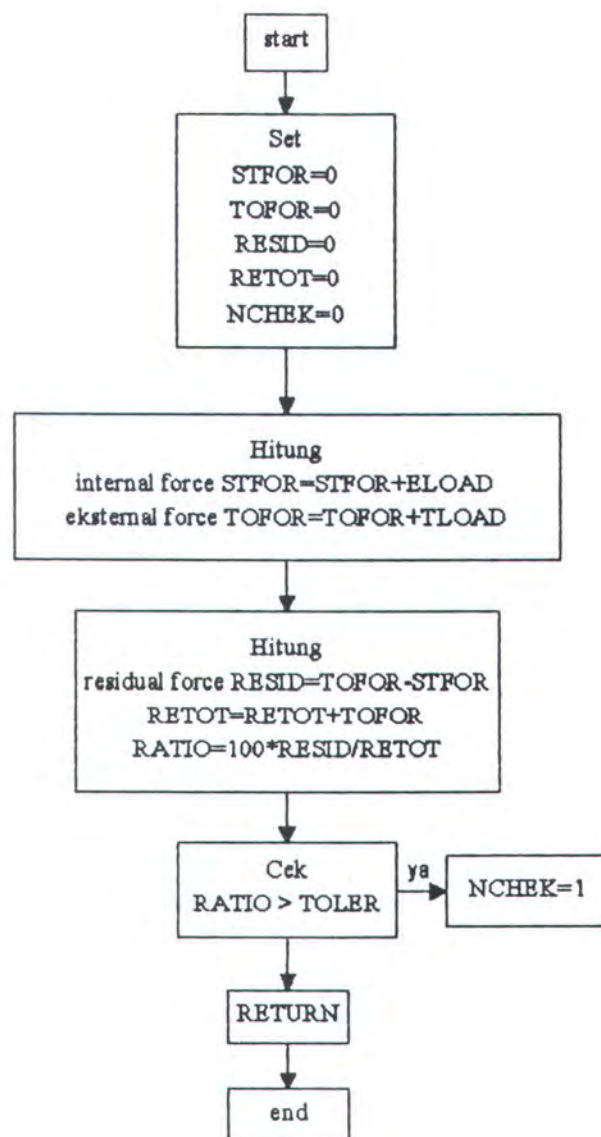
- 9 Terdapat sejumlah x kordinat titik yang sama.
- 10 Terdapat sejumlah x identifikasi material yang melebihi nilai spesifikasi total jumlah material (NMATS).
- 11 Terdapat sejumlah x nomor elemen node yang bernilai nol.
- 12 Terdapat sejumlah x nomor elemen node yang bernilai negatif atau melebihi dari nilai spesifikasi total nodal point (NPOIN).



- 13 Terdapat sejumlah x perulangan nomor elemen node dalam satu elemen.
- 14 Terdapat sejumlah x nomor nodal point yang telah dispesifikasikan tetapi tidak muncul atau terdefinisikan pada penyusunan elemen node.
- 15 Terdapat sejumlah x nomor nodal point yang tidak muncul atau terdefinisikan pada penyusunan elemen node dimana nodal point tersebut memiliki koordinat bukan nol.
- 16 Terdapat sejumlah x nomor nodal point yang tidak muncul atau terdefinisikan pada penyusunan elemen node dimana nodal point tersebut merupakan restrained nodal point.
- 17 Nilai frontwidth melebihi nilai masimum frontwidth yang telah dispesifikasikan.
- 18 Terdapat sejumlah x nomor restrained nodal point yang bernilai kurang atau sama dengan nol atau melebihi dari nilai total spesifikasi nodal point (NPOIN).
- 19 Terdapat sejumlah x restrained nodal point yang memiliki kode spesifikasi kurang atau sama dengan nol.
- 20 Terdapat sejumlah x perulangan pada penentuan restrained nodal point.

##### 5. Subroutine CONVER

Program subroutine ini berfungsi untuk memeriksa konvergensi dari proses non-linear secara iterasi. Penyelesaian yang konvergen adalah jika nilai residual force kurang dari nilai toleransi konvergensi yang telah ditetapkan. Secara garis besar proses yang terjadi pada subroutine ini adalah sebagai berikut :



Gambar 5.5. Flow chart Subroutine CONVER.

#### 6. Subroutine DIMEN

Program Subroutine ini berfungsi menetapkan nilai variabel-variabel untuk proses dimensioning yang diperlukan pada setiap program subroutine.

#### 7. Subroutine ECHO

Program subroutine ini berfungsi membaca dan menyimpan data kembali setelah pada proses program terdeteksi kesalahan data melalui subroutine CHECK1 maupun subroutine CHEK2.



#### 8. Subroutine FLOWS

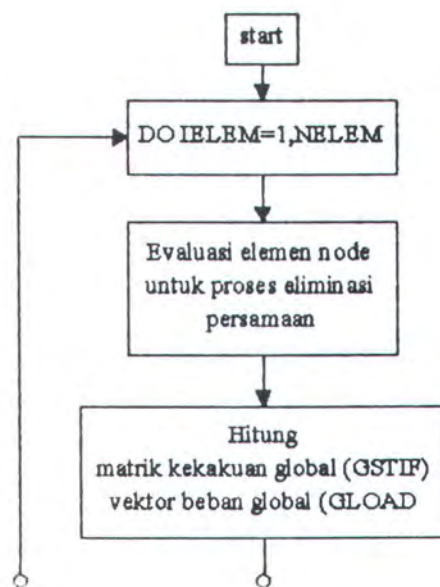
Program subroutine ini berfungsi menghitung flow vektor yang digunakan pada perhitungan plastisitas.

#### 9. Subroutine FRAME

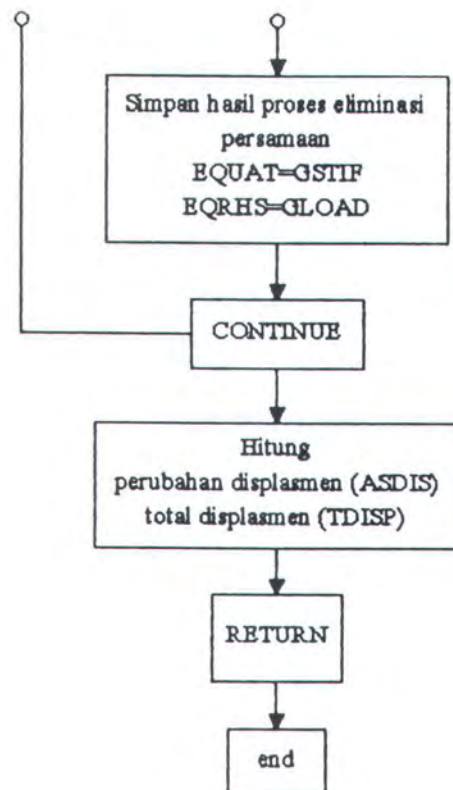
Program subroutine ini berfungsi mengendalikan bentuk matrik yang diperlukan dalam perhitungan.

#### 10. Subroutine FRONT

Program subroutine ini berfungsi membentuk matrik kekakuan global dari tiap elemen dan vektor beban global untuk digunakan pada perhitungan displasmen dimana penyelesaian perhitungan dari persamaan yang terjadi adalah menggunakan metode frontal. Secara garis besar proses yang terjadi pada subroutine adalah sebagai berikut :



Gambar 5.6. Flow chart Subroutine FRONT.



Gambar 5.7. Lanjutan *flow chart* Subroutine FRONT

#### 11. Subroutine FUNC

Program subroutine ini berfungsi untuk mengevaluasi komponen matrik B dan matrik Jacobian.

#### 12. Subroutine GAUSSQ

Program subroutine ini berfungsi menetapkan nilai faktor Gauss point untuk digunakan pada perhitungan integral.

#### 13. Subroutine GEOME

Program subroutine ini berfungsi mengevaluasi matrik kekakuan geometris untuk menghitung tegangan inisial pada saat proses pembebanan yang pertama kali.

#### 14. Subroutine INCREM

Program subroutine ini berfungsi menetapkan faktor penambahan beban yang digunakan pada proses incremental.



#### 15. Subroutine INPUT

Program subroutine ini berfungsi membaca input data untuk mendefinisikan nilai parameter data kontrol, bentuk geometris struktur (meshing), kondisi batas dan properti material struktur.

#### 16. Subroutine INVAR

Program subroutine ini berfungsi untuk memeriksa hasil tegangan dengan fungsi yield.

#### 17. Subroutine LDISP

Program subroutine ini berfungsi mengevaluasi matrik  $B_L$  yang merupakan pengaruh dari kondisi non-linear dan hasil perhitungannya ditambahkan dengan hasil perhitungan matrik  $B$ .

#### 18. Subroutine LOADS

Program subroutine ini berfungsi membaca input data beban eksternal yang diaplikasikan pada beban nodal ekuivalen.

#### 19. Subroutine MATM

Program subroutine ini berfungsi mengendalikan jenis perhitungan matrik yang dibutuhkan.

#### 20. Subroutine MODAN

Program subroutine ini berfungsi mengevaluasi matrik elastisitas (DMATT) dan matrik parameter anisotropik (AMATX).

#### 21. Subroutine NODEX

Program subroutine ini berfungsi mengevaluasi posisi node yang ada di tengah-tengah (midside node) dimana perolehannya dilakukan dengan cara interpolasi.

#### 22. Subroutine OUTPUT

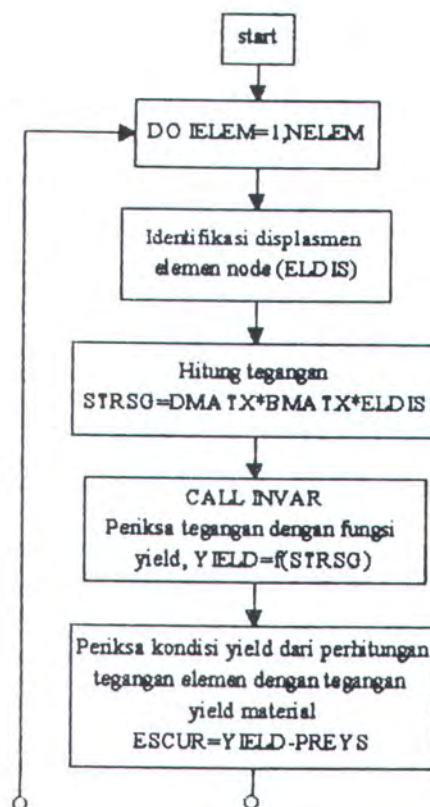
Program subroutine ini berfungsi mencetak hasil penyelesaian dari proses perhitungan displasmen, gaya reaksi dan tegangan.

### 23. Subroutine PRES

Program subroutine ini berfungsi mengevaluasi jenis aplikasi pembebanan tekanan (pressure).

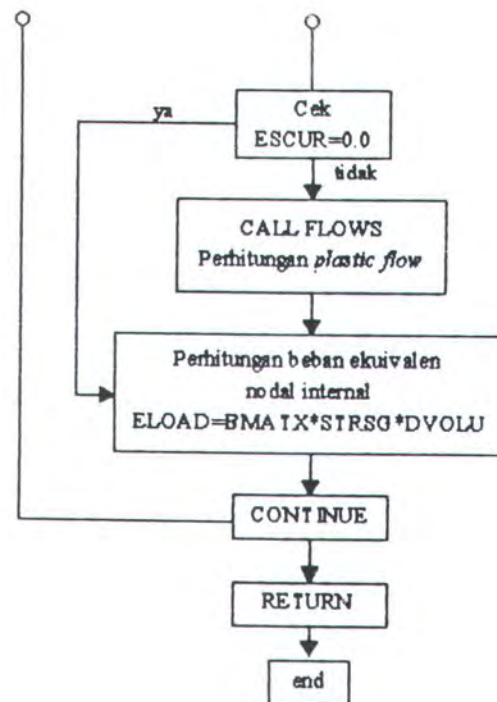
### 24. Subroutine RESTR

Program subroutine ini berfungsi mengevaluasi tegangan dan beban nodal ekuivalen yang disesuaikan dengan kondisi yield saat itu. Secara garis besar proses yang terjadi pada subroutine ini adalah sebagai berikut :



Gambar 5.8. Flow chart Subroutine RESTR.





Gambar 5.9. Lanjutan *flow chart* Subroutine RESTR.

#### 25. Subroutine SFR1

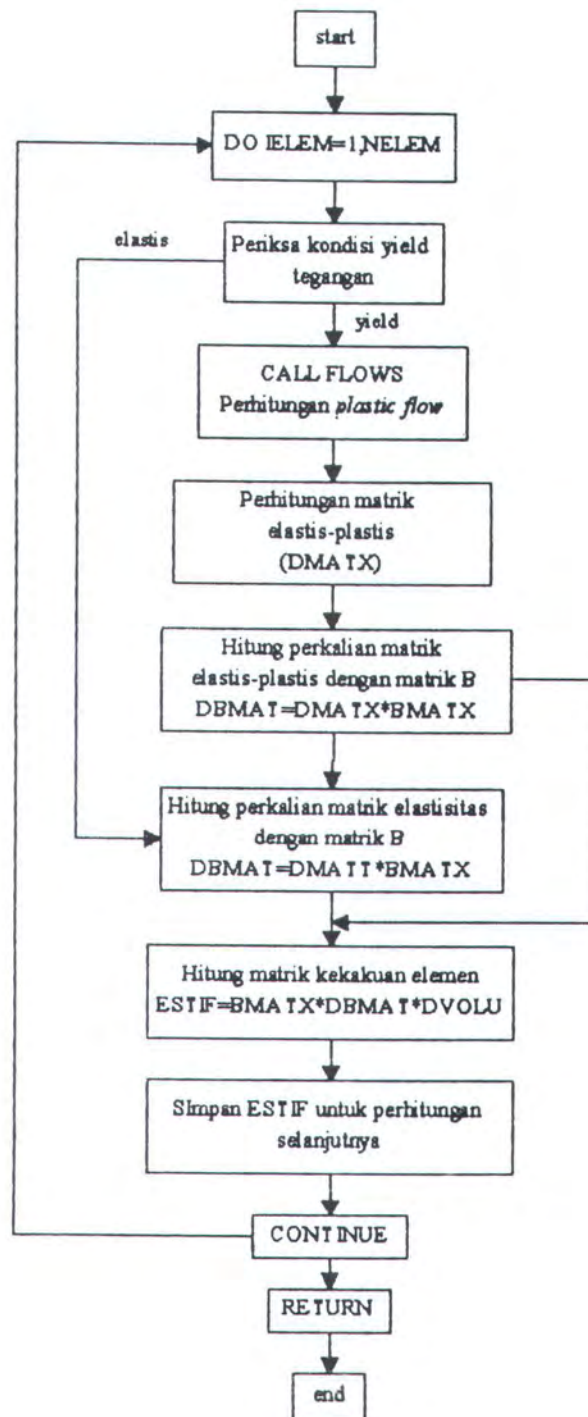
Program subroutine ini berfungsi mengevaluasi nilai shape function dan nilai derivatifnya.

#### 26. Subroutine SINGOP

Program subroutine ini berfungsi mengendalikan bentuk matrik yang dibutuhkan.

#### 27. Subroutine STIFF

Program subroutine ini berfungsi mengevaluasi matrik kekakuan dari tiap elemen. Secara garis besar proses yang terjadi pada subroutine ini adalah sebagai berikut :



Gambar 5.10. Flow chart Subroutine STIFF.



#### 28. Subroutine VECT

Program subroutine ini berfungsi mengendalikan jenis perhitungan matrik yang dibutuhkan.

#### 29. Subroutine WORKS

Program subroutine ini berfungsi menentukan sistem axis dari koordinat dan mengevaluasi ketebalan elemen.

#### 30. Subroutine RESTAR

Program subroutine ini berfungsi menyimpan hasil penyelesaian dari proses incremental dimana proses penyelesaiannya telah mencapai konvergen.

#### 31. Subroutine ZERO

Program subroutine ini berfungsi memberikan nilai inisial nol pada beberapa variabel untuk digunakan pada proses akumulasi atau penjumlahan nilai yang baru dari hasil proses incremental maupun proses iterasi.

#### 32. Subroutine SHEARC

Program subroutine ini berfungsi menetapkan nilai faktor shear correction untuk kasus struktur laminasi.

### 5.4. Variabel Input Data

Input data pada program analisa elastis-plastis ini terdiri dari delapan macam data yang diantaranya merupakan pendefinisian dari bentuk geometris (mesh) dari struktur, properti material, kondisi batas dan jenis pembebanan. Kedelapan macam data tersebut adalah sebagai berikut :

#### 1. Judul data :

TITLE

#### 2. Kontrol data :

NPOIN Total jumlah nodal poin.

NELEM Total jumlah elemen.

NVFIX Total jumlah titik terkekang (restrained node).

NNODE Total jumlah node per elemen.

= 8

= 9

NMATS Total jumlah jenis material.

NGAUS Jumlah Gauss point per elemen.

NGAUZ Jumlah Gauss point per elemen.

NGAUS = 3, NGAUZ = 3 - *Normal integration rule.*

NGAUS = 3, NGAUZ = 2 - *Selective integration rule.*

NGAUS = 2, NGAUZ = 2 - *Reduced integration rule.*

NCOLA Set konstrain dari elemen 9 node.

= 0 no konstrain.

= 1 konstrain (u,v,w).

NALGO Indikator proses non-linear.

= 1 perhitungan matrik kekakuan hanya dilakukan pada proses incremental yang pertama kali (*initial stiffness*).

= 2 perhitungan matrik kekakuan dilakukan pada setiap proses incremental maupun proses iterasi (*tangential stiffness*).

= 3 Perhitungan matrik kekakuan hanya dilakukan pada proses iterasi yang pertama kali untuk setiap proses incremental.

= 4 perhitungan matrik kekakuan hanya dilakukan pada proses iterasi yang kedua untuk setiap proses incremental.

NINCS Total jumlah proses incremental.

NLAYR Total jumlah layer untuk kasus struktur laminasi.

LARGE Parameter analisa deformasi.

= 0 analisa linear geometris.

= 1 analisa non-linear geometris.



NREST Parameter restart analisa.

= 0 start analisa.

= 1 restart analisa dari hasil penyelesaian konvergensi pada proses incremental.

3. Data percepatan gravitasi :

GRAVI(1) Percepatan gravitasi arah x.

GRAVI(2) Percepatan gravitasi arah y.

GRAVI(3) Percepatan gravitasi arah z.

ANVEL Kecepatan angular.

4. Elemen data :

NUMEL Nomor elemen.

MATNO((NUMEL,ILAYR),ILAYR=1,NLAYR) Nomor material.

LNODS(NUMEL,INODE),INODE=1,NNODE) Nomor node elemen.

5. Koordinat data :

IPOIN Nomor nodal point.

COORD(IPOIN,1) Top koordinat x.

COORD(IPOIN,2) Top koordinat y.

COORD(IPOIN,3) Top koordinat z.

COORD(IPOIN,4) Top pressure.

COORD(IPOIN,5) Bottom koordinat x.

COORD(IPOIN,6) Bottom koordinat y.

COORD(IPOIN,7) Bottom koordinat z.

COORD(IPOIN,8) Bottom pressure.

6. Restrain data :

NOFIX Nomor retrained node.

IFPRE Kondisi derajat kebebasan (u,v,w, $\beta_1$ , $\beta_2$ ).

= 0 bebas.

= 1 terkekang.

7. Material data :

- PROPS(NUMAT,1)  $E_1$  Modulus Young (x-direction).  
 PROPS(NUMAT,2)  $\nu$  Poisson ratio.  
 PROPS(NUMAT,3)  $t$  Tebal layer.  
 PROPS(NUMAT,4)  $\rho$  Densitas material.  
 PROPS(NUMAT,5)  $\alpha$  Koefisien panas.  
 PROPS(NUMAT,6)  $\sigma_{01}$  Tegangan yield (x-direction).  
 PROPS(NUMAT,7)  $H$  Parameter hardening.  
 PROPS(NUMAT,8)  $E_2$  Modulus Young (y-direction).  
 PROPS(NUMAT,9)  $G_{12}$  Modulus shear (xy-plane).  
 PROPS(NUMAT,10)  $G_{13}$  Modulus shear (xz-plane).  
 PROPS(NUMAT,11)  $G_{23}$  Modulus shear (yz-plane).  
 PROPS(NUMAT,12)  $\sigma_{02}$  Tegangan yield (y-direction).  
 PROPS(NUMAT,13)  $\sigma_{03}$  Tegangan yield (z-direction).  
 PROPS(NUMAT,14)  $\tau_{12}$  Tegangan shear yield (xy-plane).  
 PROPS(NUMAT,15)  $\tau_{13}$  Tegangan shear yield (xz-plane).  
 PROPS(NUMAT,16)  $\tau_{23}$  Tegangan shear yield (yz-plane).  
 PROPS(NUMAT,17)  $\theta$  Sudut layer pada material.

8. Data beban :

- NPRES Indikator untuk beban terdistribusi.  
     = 0 pada elemen tidak terdapat beban terdistribusi.  
     = 1 pada elemen terdapat beban terdistribusi.  
 NUCLO Jumlah beban terkonsentrasi.  
 NBODY Indikator untuk beban akibat berat struktur .  
     = 0 terjadi pada elemen.  
     = 1 tidak terjadi pada elemen.



Jika NPRES = 1,

KPRES Indikator jenis beban terdistribusi.

= 0 beban terdistribusi uniform.

= 1 beban hidrostatik.

= 2 beban dispesifikasikan sebagai beban nodal seperti pada koordinat data sebelumnya (top dan bottom pressure).

CFACE = +1 top pressure.

= -1 bottom pressure.

PREVA Nilai beban terdistribusi untuk KPRES = 0 atau KPRES = 1.

SURFA Koordinat z untuk zero pressure jika KPRES = 1.

Jika NUCLO > 0,

LPOIN Nomor lokal elemen node (1-8) dimana beban terkonsentrasi.

LDOFN Arah beban konsentrasi bekerja.

= 1 x-displasmen.

= 2 y-displasmen.

= 3 z-displasmen.

= 4 rotasi  $\beta_1$ .

= 5 rotasi  $\beta_2$ .

CARGA Nilai beban terkonsentrasi.

#### 9. Kontrol incremental data :

FACTO Faktor penambahan beban.

TOLER Faktor toleransi konvergen.

MITER Maksimum iterasi.

NOUTP (1) Parameter kontrol untuk output hasil unkonvergen setelah iterasi pertama.

= 1 Cetak hasil displasmen.

= 2 Cetak hasil displasmen dan gaya reaksi.

= 3 Cetak hasil displasmen, gaya reaksi dan tegangan.

NOUTP (2) Parameter kontrol untuk output hasil konvergensi.

= 1 Cetak hasil displasmen.

= 2 Cetak hasil displasmen dan gaya reaksi simpul terkekang.

= 3 Cetak hasil displasmen, gaya reaksi simpul terkekang dan tegangan.

### 5.5. Variabel-variabel Program

Berikut di bawah ini dijelaskan beberapa variabel program lainnya yang dianggap penting, yaitu :

ASDIS	Perubahan displasmen.
ELOAD	Beban ekuivalen.
EQRHS	Variabel penyimpan hasil persamaan matrik vektor pembebanan pada proses frontal.
EQUAT	Vaiabel penyimpan hasil persamaan matrik kekakuan global pada proses frontal.
EFFST	Tegangan efektif pada kondisi yield.
EPSTN	Regangan efektif pada kondisi plastis.
ESTIF	Matrik kekakuan lokal elemen.
GLOAD	Matrik vektor pembebanan.
GSTIF	Matrik kekakuan global.
IFFIX	Kode simpul terkekang.
LOCEL	Kondisi lokal elemen pada proses frontal/reduksi <i>d.o.f</i> .
NACVA	Jumlah variabel aktif pada proses frontal.
POSGP	Konstanta <i>Gauss</i> .
WEIGP	Konstanta <i>Gauss</i> .
STRSG	Tegangan elemen.



TDISP	Total displasmen.
TLOAD	Total beban eksternal.
TREAC	Gaya reaksi pada simpul terkekang.
VECRV	Variabel penyimpanan hasil penyelesaian persamaan displasmen pada proses frontal.

### 5.6. Program Analisa Grafis

Program ini adalah merupakan satu program tambahan yang disusun tersendiri untuk melihat kondisi model struktur sebelum maupun sesudah deformasi. Pada penyusunannya, program memanfaatkan fasilitas pustaka grafis (graphics.lib) dari program MS FORTRAN versi 5.10.

Input data untuk program tambahan ini berasal dari hasil penyimpanan data pada program utama yang merupakan data-data dari jumlah koordinat (NPOIN), jumlah elemen (NELEM), koordinat awal (COORD), penomoran elemen (LNODS) dan nilai total displasmen (TDISP) pada akhir proses incremental yang ditetapkan.





# **TUGAS AKHIR**

## **BAB VI**

### **ANALISA HASIL PROGRAM**



## BAB VI

### ANALISA HASIL PROGRAM

#### 6.1. Analisa Hasil dan Pembahasan

Aplikasi program analisa permasalahan non-linear seperti yang dijelaskan pada bab sebelumnya, selanjutnya akan digunakan untuk memprediksi secara numerik kapasitas kekuatan *ultimate* dari sambungan IT dan PT *joint*. Beberapa kondisi batas dan perlakuan diberikan pada input program untuk efektifitas dan efisiensi saat proses program, sedangkan verifikasi dari hasil proses program analisa permasalahan non-linear dilakukan dengan membandingkan hasil program dengan hasil pengujian maupun hasil penurunan rumus dari tabel data referensi, untuk permasalahan perhitungan tipe sambungan dan kondisi pembebanan yang sama.

##### 6.1.1. Pemodelan Elemen

Suatu analisa mekanika struktur dengan pendekatan metode elemen hingga dilakukan dengan cara pendiskritan suatu bentuk yang kontinu menjadi bentuk model yang tersusun dari beberapa bagian elemen. Proses perhitungan yang akan terjadi adalah perhitungan secara model matematis yang harus dapat mewakili dari sistem struktur yang sebenarnya, dan hasil perhitungan yang diperoleh tidaklah terlalu eksak. Tingkat keakurasian hasil yang diharapkan sangat tergantung dari tingkat kesalahan numerik yang terjadi pada proses perhitungan dan kondisi pemodelan elemen yang ditetapkan seperti pemilihan tipe, jumlah maupun bentuk penyusunan elemen yang tepat dan juga pemberian kondisi batas pada model elemen yang

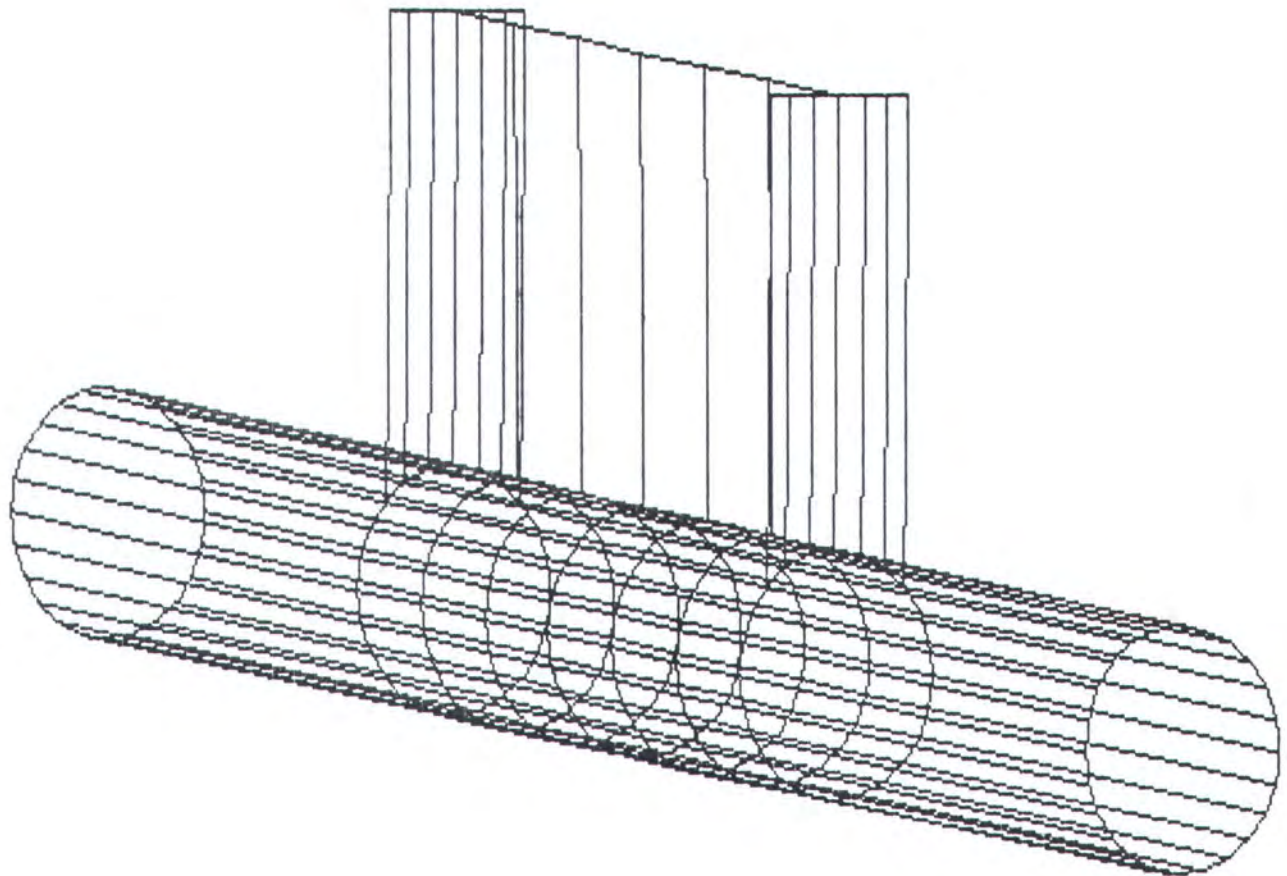
sedemikian rupa hingga dapat mendekati dengan kondisi fisik struktur yang sebenarnya.

Pada pembahasan tugas akhir ini, penyusunan model elemen (*meshing*) dilakukan berdasarkan pertimbangan-pertimbangan tertentu dimana tanpa mengabaikan perhatian terhadap tingkat keakurasian terhadap hasil yang diharapkan, pertimbangan-pertimbangan tersebut dapat merupakan suatu batasan maupun kendala dalam penyusunan model elemen. Pertimbangan-pertimbangan dan kondisi batas yang diberikan kepada elemen pada permasalahan tugas akhir ini adalah sebagai berikut :

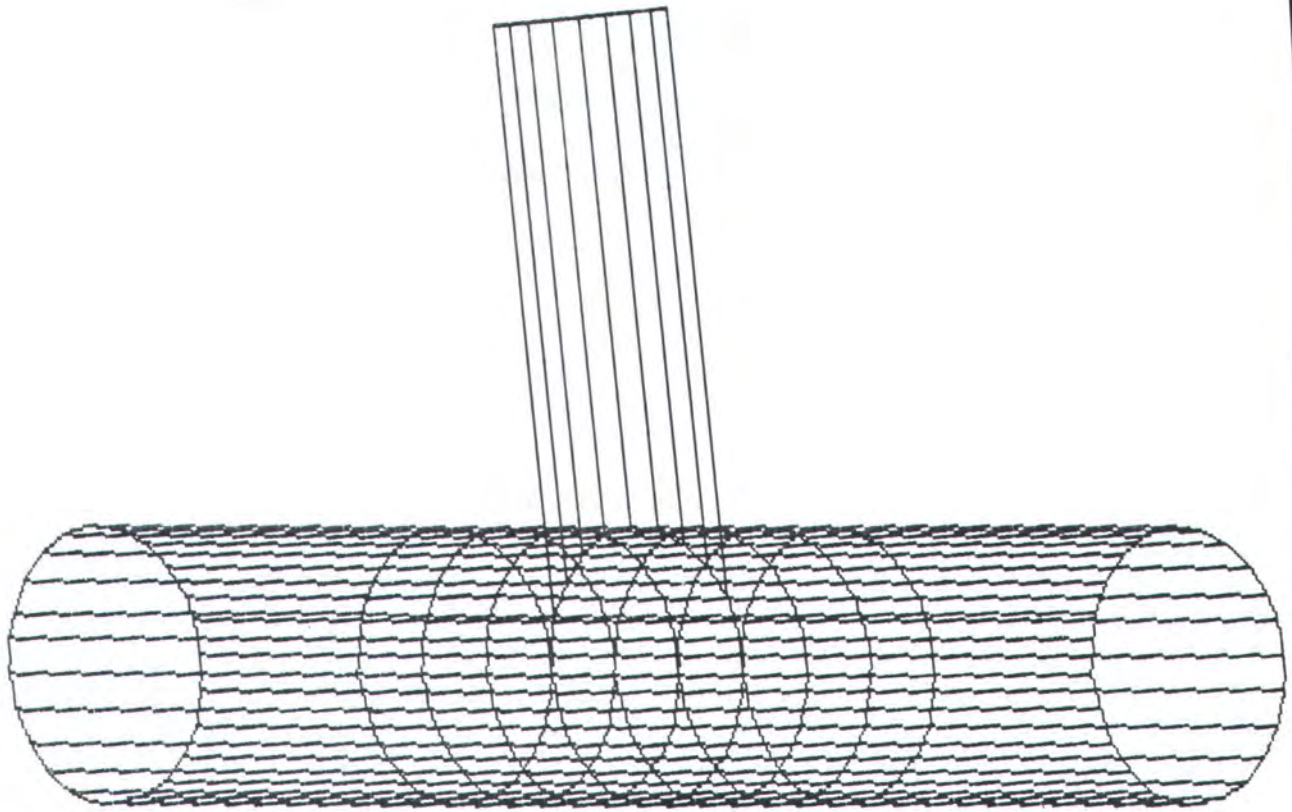
1. Tinjauan masalah yang ada, dimana pada permasalahan tugas akhir ini pokok permasalahan dianggap sebagai permasalahan non-linear. Pada analisa permasalahan non-linear, untuk penyelesaian persamaan-persamaan yang terjadi dengan sejumlah derajat kebebasan yang sama (*d.o.f.*) dapat memberikan biaya komputasi 10 hingga 100 kali lebih besar dibanding dengan penyelesaian persamaan pada permasalahan linear [Ref. 3]. Hal tersebut menimbulkan kendala bagi tingkat kompleksitas dari model elemen yang diinginkan. Kendala ini pada permasalahan tugas akhir akan menyangkut terbatasnya jumlah elemen yang dapat digunakan.
2. Tujuan analisa pada permasalahan tugas akhir, adalah untuk mengetahui seberapa jauh kapasitas kekuatan *ultimate* sambungan terhadap total beban eksternal yang saat itu bekerja pada sambungan ( $P_{ult}$ ). Analisa tersebut pada dasarnya adalah berdasarkan pada penyelesaian konvergensi nilai *residual force* atau keseimbangan gaya eksternal dan internal yang terjadi pada saat itu, jadi dalam hal ini analisa konsentrasi tegangan diabaikan. Batasan ini dimaksudkan untuk menghindari proses *meshing refinement* atau penghalusan penyusunan elemen pada daerah lokal tertentu untuk mendapatkan tegangan maksimum yang lebih baik, sehingga dalam hal ini akan terjadi proses



penambahan elemen yang menyebabkan bertambahnya kebutuhan kapasitas memori program. Contoh model penyusunan elemen yang digunakan pada analisa tugas akhir adalah seperti yang terlihat pada Gambar 6.1. dan 6.2.



Gambar 6.1. Model elemen IT *Joint*.

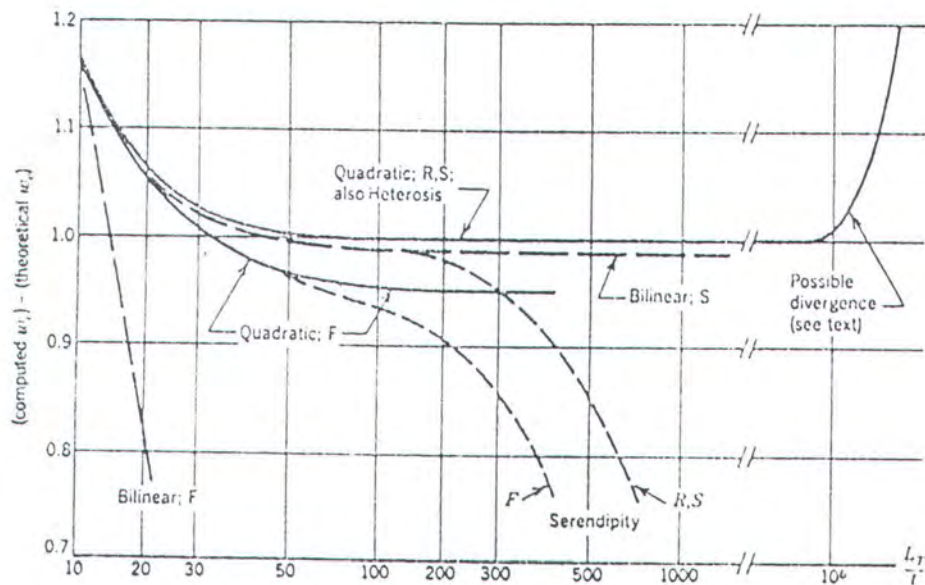


Gambar 6.2. Model elemen PT Joint.

3. Tipe atau jenis elemen yang digunakan adalah elemen *shell serendipity* dengan 8 nodal poin (40 *d.o.f.*). Pemilihan tipe elemen *shell* disesuaikan dengan aplikasi input program yang digunakan pada analisa tugas akhir, sedangkan penyelesaian integrasi digunakan aturan *low-order Gauss Quadrature (2×2)* atau tipe integrasi *reduced integration*. Pemilihan jenis integrasi didasarkan pada tipe elemen, jumlah nodal poin tiap elemen dan kondisi perbandingan sisi panjang elemen ( $L_T$ ) dengan ketebalan elemen ( $t$ ) yang nilainya terletak antara  $33.33 < L_T/t < 57.69$ . Pada Gambar grafik 6.3. terlihat bahwa untuk tipe elemen



dan kondisi  $L_T/t$  yang ada, penyelesaian integrasi dengan *reduced integration* masih memberikan hasil yang lebih baik.



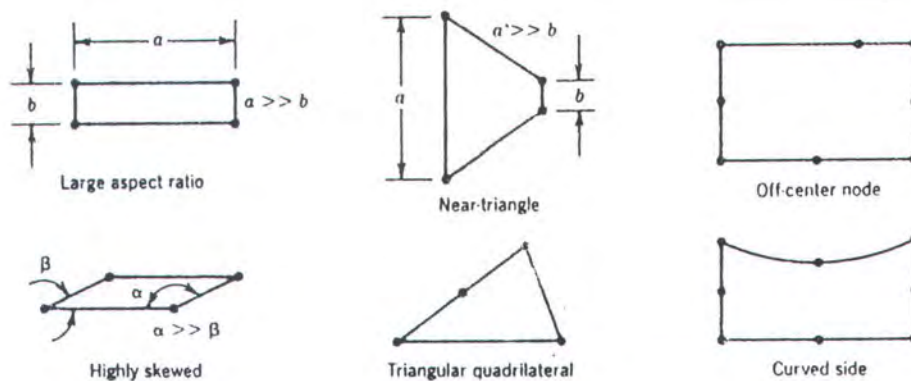
Gambar 6.3. Pemilihan jenis integrasi *reduced* (R), *selective* (S) dan *full* (F) menurut jenis elemen.

4. Dimensi model elemen, sebagai verifikasi dan perbandingan hasil maka dimensi model elemen disesuaikan dengan referensi data dimensi dari model pengujian.

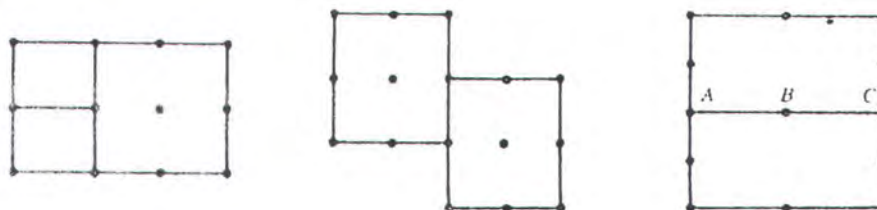
Dimensi tersebut meliputi :

- Panjang tubular ( $L_0$ )
- Diameter tubular (D)
- Tebal tubular (T)
- Panjang beam/pelat ( $L_1$ )
- Tinggi beam (h)
- Lebar beam/pelat (b)
- Tebal *flange* (tf)
- Tebal *web* (tw)

5. Kondisi batas untuk sistem sambungan adalah dikenakan pada kedua ujung *chord* yang dianggap sebagai tumpuan pin dan untuk tiap nodal poin elemen lainnya masing-masing memiliki 5 derajat kebebasan (*d.o.f.*) yaitu (3 arah translasi dan 2 rotasi). Khusus untuk masalah ini, untuk melihat seberapa jauh penyimpangan dari hasil yang diharapkan maka sebelumnya dapat dibandingkan dari bentuk deformasi yang terjadi pada sambungan terhadap estimasi model deformasi sambungan yang akan terjadi akibat proses pembebanan yang ada.
6. Menghindari bentuk-bentuk elemen dan hubungan nodal poin antar elemen yang dapat mengurangi keakurasian hasil perhitungan, seperti yang terlihat pada Gambar 6.4. dan 6.5.



Gambar 6.4. Bentuk elemen yang tidak baik.



Gambar 6.5. Bentuk hubungan nodal point antar elemen yang tidak baik.



### 6.1.2. Data Geometri Elemen

Sebagai verifikasi dan perbandingan hasil analisa program maka beberapa input data geometri pada model elemen yang akan dianalisa adalah mengambil dari tabel data pada referensi [1] yang merupakan data hasil pengujian sambungan IT dan PT *joint* dan data perbandingannya dengan hasil penurunan rumus. Sedangkan parameter-parameter geometri yang dianggap mempengaruhi pada saat analisa adalah seperti yang dijelaskan pada bab pendahuluan yaitu  $\gamma$  ( $D/2T$ ),  $\beta$  ( $b/D$ ) dan  $\eta$  ( $h/D$ ) dan sebagai batasan masalah dalam tugas akhir ini perubahan parameter yang ditinjau hanyalah perubahan parameter  $\beta$ .

Dalam bentuk tabel, data geometri pada model elemen yang akan dianalisa atau digunakan sebagai input program adalah sebagai berikut :

Tabel 6.1. Data geometri model elemen.

type	D	T	b	h	$L_n$	$L_1$	tf	tw	$\gamma$	$\beta$	$\eta$
<i>joint</i>	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	D/2T	b/D	h/D
IT	165.2	5.2	82.6	330.4	930.4	300	9	5.8	15.9	0.5	2.0
IT	165.2	5.2	99.12	330.4	930.4	300	9	5.8	15.9	0.6	2.0
IT	165.2	5.2	115.6	330.4	930.4	300	9	5.8	15.9	0.7	2.0
IT	165.2	5.2	132.2	330.4	930.4	300	9	5.8	15.9	0.8	2.0
IT	165.2	5.2	148.7	330.4	930.4	300	9	5.8	15.9	0.9	2.0
PT	165.2	5.2	82.6	-	930.4	300	9	-	15.9	0.5	-
PT	165.2	5.2	99.12	-	930.4	300	9	-	15.9	0.6	-
PT	165.2	5.2	115.6	-	930.4	300	9	-	15.9	0.7	-
PT	165.2	5.2	132.2	-	930.4	300	9	-	15.9	0.8	-
PT	165.2	5.2	148.7	-	930.4	300	9	-	15.9	0.9	-

### 6.1.3. Data Material

Jenis ataupun kondisi material pada elemen yang digunakan sebagai input program adalah dianggap sama atau homogen untuk seluruh bagian elemen sambungan, dalam hal ini bagian elemen yang mendukung sambungan adalah elemen tubular sebagai bagian utama sambungan (*chord*) dan bagian sambungan lainnya (*brace*) adalah elemen *I-beam* atau elemen *plate*. Sebagai perbandingan dengan hasil data pengujian yang ada maka input kondisi material yang harus disesuaikan adalah kondisi tegangan yield material. Secara lengkap input program berupa data material adalah sebagai berikut :

Modulus Young (E)	:	200000 N/mm <sup>2</sup>
Modulus shear (G)	:	84000 N/mm <sup>2</sup>
Poisson ratio ( $\nu$ )	:	0.3
Material density ( $\rho$ )	:	-
Koefisien thermal ( $\alpha$ )	:	-
Uniaxial yield stress, $\sigma_x$	:	308.0 N/mm <sup>2</sup>
	$\sigma_y$	: 308.0 N/mm <sup>2</sup>
	$\sigma_z$	: 308.0 N/mm <sup>2</sup>
Shear yield stress, $\tau_{xy}$	:	185.176 N/mm <sup>2</sup>
	$\tau_{xz}$	: 185.176 N/mm <sup>2</sup>
	$\tau_{yz}$	: 185.176 N/mm <sup>2</sup>

### 6.1.4. Kondisi Pembebanan

Sebagai batasan masalah untuk kondisi pembebanan pada permasalahan tugas akhir ini, hanya ditinjau kondisi pembebanan berupa beban aksial tekan (*compression*). Karena untuk memprediksi nilai kapasitas kekuatan *ultimate* sambungan pada program dilakukan secara incremental dan iterasi maka diperlukan



beberapa langkah perlakuan dan pengamatan untuk mengefisiensikan waktu proses program, yaitu sebagai berikut :

1. Total input beban eksternal ( $P_{tot}$ ) yang diberikan pada model struktur adalah diusahakan sama atau mendekati dengan nilai beban *ultimate* ( $P_{ult}$ ) yang dapat diperoleh dari referensi data hasil pengujian maupun hasil penurunan rumus.
2. Pada setiap proses incremental, total beban eksternal ( $P_{eks}$ ) yang bekerja pada saat itu adalah sama dengan faktor incremental pembebanan ( $f_i$ ) pada saat itu dikalikan dengan total input beban eksternal ( $P_{tot}$ ) yang sebelumnya sudah ditetapkan, dalam suatu persamaan, pernyataan tersebut di atas dapat dijelaskan secara umum sebagai berikut :

$$P_{eks} = (f_1 + f_2 + f_3 + \dots + f_n) P_{tot} \quad (6-1.1)$$

atau dapat ditulis menjadi,

$$P_{eks} = P_{tot} \left( \sum_{i=1, n}^n f_i \right) \quad (6-1.2)$$

dimana,

$P_{eks}$  = total beban eksternal yang bekerja pada proses incremental pembebanan saat itu

$P_{tot}$  = total beban eksternal yang telah ditetapkan pada input program sebelumnya

$f_i$  = faktor incremental pembebanan

$n$  = proses incremental yang ke-n

3. Selanjutnya pengamatan penyelesaian konvergensi dari program dilakukan. Jika penyelesaian konvergensi sulit dicapai hingga batas maksimum iterasi maka hal ini dapat dianggap bahwa beban eksternal yang bekerja saat itu adalah melebihi beban *collapse* atau beban *ultimate* pada model struktur ( $P_{ult}$ ). Selanjutnya

usaha pendekatan untuk mencari nilai beban *ultimate* yang sebenarnya adalah dilakukan dengan mengubah atau mengurangi nilai faktor incremental pembebanan ( $f_i$ ) pada proses incremental dimana saat itu penyelesaian konvergensi sulit dicapai. Sebagai catatan bahwa jika pada program, penyelesaian konvergensi sulit dicapai maka output hasil penyelesaian yang terjadi seperti displasmen dan tegangan tidak akan pernah disimpan atau akan diabaikan untuk proses incremental selanjutnya jadi output hasil penyelesaian hanya akan disimpan jika proses konvergensi terjadi pada program.

4. Khusus untuk tugas akhir ini pada permasalahan efisiensi waktu proses program, pemberian nilai faktor incremental pembebanan ( $f_i$ ) untuk pertama kalinya sebelum proses *un-konvergensi* yang pertama terjadi adalah 0.1 untuk setiap proses incremental. Selanjutnya pengurangan nilai faktor incremental pembebanan pada proses incremental berikutnya adalah dengan memasukkan nilai  $f_i$  yang baru yaitu 0.01 untuk setiap proses incremental. Identik dengan sebelumnya, setelah proses *un-konvergensi* yang kedua terjadi maka pengurangan nilai faktor incremental pembebanan berikutnya menjadi 0.001. Pada tugas akhir ini, pendekatan nilai faktor incremental pembebanan  $f_i$  adalah hanya sampai tiga angka dibelakang koma.

Berikut ini adalah satu contoh penjelasan dari pembahasan di atas :

- Diketahui bahwa menurut data pengujian kapasitas *ultimate* dari tipe sambungan IT *joint* adalah 260.5 kN. Selanjutnya dengan menggunakan suatu model elemen struktur, kapasitas *ultimate* dari tipe sambungan tersebut akan didekati dengan aplikasi program non-linear dan langkah-langkah yang dilakukan pada proses program adalah sebagai berikut :

1. Memasukkan total input beban eksternal ( $P_{tot}$ ) yang sama dengan nilai kapasitas *ultimate* sambungan menurut data pengujian atau  $P_{tot} = 260.5$  kN.



2. Memberi nilai faktor incremental pembebanan ( $f_i$ ) untuk pertama kalinya sebesar 0.1 pada setiap proses incremental.
3. Melakukan pengamatan proses konvergensi, misalnya pada proses incremental pembebanan yang ke 10 dimana saat itu proses konvergensi gagal dicapai hingga batas maksimum iterasi maka sesuai dengan persamaan 6.1.1. dan 6.1.2. total beban eksternal pada saat itu adalah :

$$\begin{aligned}
 P_{eks} &= P_{tot} \times (10 \times 0.1) \\
 &= 260.5 \times 1.0 \\
 &= 260.5 \text{ kN}
 \end{aligned}$$

sehingga dapat diprediksi bahwa nilai  $P_{ult}$  adalah terletak di antara total beban eksternal pada proses incremental yang ke 9 dan ke 10, yaitu :

$$234.45 \text{ kN} < P_{ult} < 260.5 \text{ kN}$$

4. Mengurangi nilai faktor incremental pembebanan menjadi 0.01 pada setiap proses incremental.
5. Melakukan pengamatan proses konvergensi yang kedua kalinya, misalnya pada tujuh proses incremental berikutnya proses *un*-konvergenensi terjadi maka diperkirakan nilai  $P_{ult}$  saat itu terletak pada :

$$\begin{aligned}
 260.5 \times (9 \times 0.1 + 6 \times 0.01) &< P_{ult} < 260.5 \times (9 \times 0.1 + 7 \times 0.01) \\
 250.08 \text{ kN} &< P_{ult} < 252.685
 \end{aligned}$$

6. Memberi nilai faktor incremental pembebanan yang baru menjadi 0.001 pada setiap proses incremental.

7. Melakukan pengamatan proses konvergensi yang terakhir, misalnya pada lima proses incremental berikutnya proses *un*-konvergensi terjadi maka dapat diprediksi bahwa nilai  $P_{ult}$  untuk model struktur yang dianalisa adalah :

$$P_{ult} \approx 260.5 \times (9 \times 0.01 + 6 \times 0.01 + 4 \times 0.001)$$

$$P_{ult} \approx 251.122 \text{ kN}$$

#### 6.1.5. Perbandingan Hasil Nilai Kapasitas *Ultimate*

Sebagai verifikasi dari hasil program analisa permasalahan non-linear, berikut ini adalah tabel perbandingan nilai  $P_{ult}$  antara hasil program dengan data referensi hasil pengujian maupun hasil dari penurunan rumus :

Tabel 6.2.  $P_{ultimate}$  dari hasil pengujian, penurunan rumus dan hasil analisa program.

tipe <i>joint</i>	teg. yield N/m <sup>2</sup>	kond. beban	$\gamma$ D/2 T	$\beta$ b/D	$\eta$ h/D	Pult test kN	Pult Kuro. kN	Pult Ward. kN	Pult EPR kN	Pult prog. kN
IT	308.	tekan	15.9	0.7	2.0	260.5	260.5	182.2	192.6	251.1
PT	308.	tekan	15.9	0.7	-	171.2	174.7	121.4	128.7	170.3

dari tabel perbandingan di atas, selanjutnya untuk verifikasi hasil penyelesaian program analisa permasalahan non-linear adalah dengan membandingkan nilai

$\frac{P_{ult}(\text{test})}{P_{ult}(\text{program})}$  dimana dalam bentuk tabel hasil perbandingan antara  $P_{ultimate}$

hasil test dengan keseluruhan hasil penurunan rumus maupun hasil penyelesaian program adalah seperti berikut ini :



Tabel 6.3. Perbandingan  $P_{ultimate}$  pengujian dengan hasil  $P_{ultimate}$  penurunan rumus dan hasil analisa program.

tipe joint	$\frac{P_{ult}(test)}{P_{ult}(Kuro)}$	$\frac{P_{ult}(test)}{P_{ult}(Ward.)}$	$\frac{P_{ult}(test)}{P_{ult}(EPR)}$	$\frac{P_{ult}(test)}{P_{ult}(program)}$
IT	1.00	1.43	1.35	1.04
PT	0.98	1.41	1.33	1.01

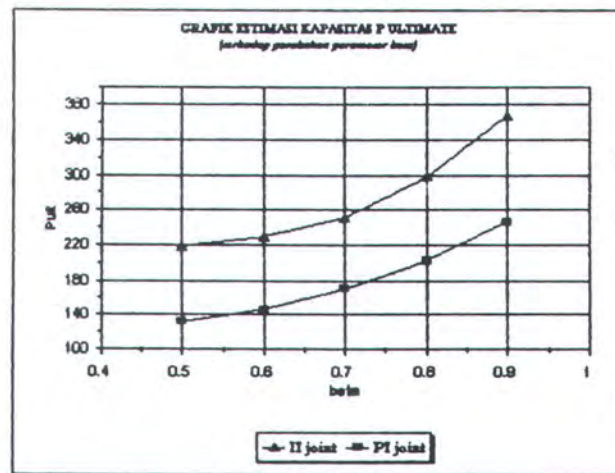
dengan hasil tabel perbandingan di atas dapat disimpulkan bahwa hasil penyelesaian program dan hasil penurunan persamaan Kurobane lebih mendekati dengan hasil pengujian.

#### 6.1.6. Analisa Pengaruh Perubahan Parameter $\beta$

Hubungan antara nilai kapasitas *ultimate* beban pada masing-masing tipe sambungan akibat pengaruh perubahan parameter  $\beta$  dari hasil penyelesaian program analisa permasalahan non-linear, adalah seperti yang tercantum pada tabel berikut ini :

Tabel 6.4. Estimasi $P_{ultimate}$ dengan perubahan parameter $\beta$ [kN]		
$\beta$	IT joint	PT joint
0.5	218.56	130.28
0.6	228.72	146.55
0.7	251.12	170.34
0.8	298.53	202.70
0.9	367.83	245.84

sedangkan dalam bentuk grafik hubungan tersebut di atas dapat dilihat seperti pada Gambar 6.6. berikut ini :



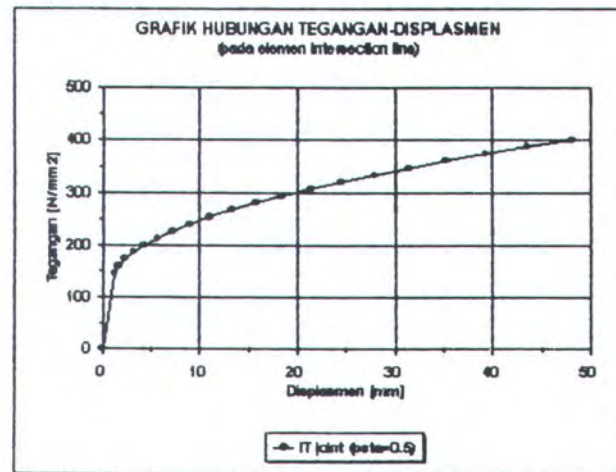
Gambar 6.6. Grafik Estimasi Kapasitas  $P$  Ultimate

Interpretasi dari Gambar grafik di atas dapat terlihat bahwa hubungan nilai kapasitas beban ( $P$ ) *ultimate* pada masing-masing tipe sambungan IT dan PT *joint* adalah konsisten secara non-linear terhadap perubahan parameter  $\beta$  dan nilai kapasitas  $P$  *ultimate* pada tipe sambungan IT *joint* adalah relatif lebih besar dibandingkan dengan nilai kapasitas  $P$  *ultimate* pada tipe sambungan PT *joint* untuk kondisi parameter yang sama.

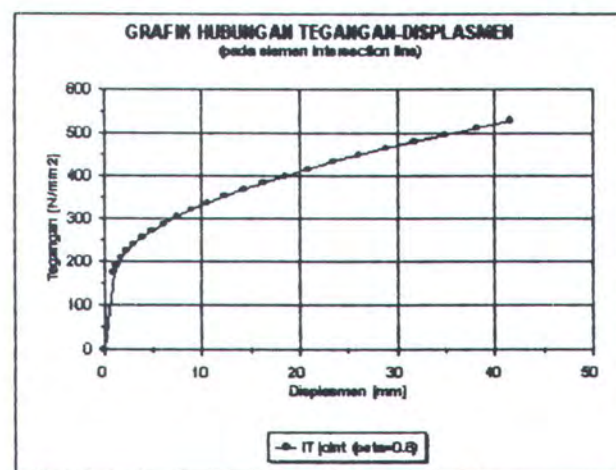
#### 6.1.7. Hubungan Tegangan dan Deformasi

Hubungan antara tegangan dan deformasi pada masing-masing tipe sambungan IT dan PT *joint* dengan parameter  $\beta$  yang berbeda dapat dilihat pada Gambar-gambar grafik dibawah ini, dimana deformasi dalam hal ini diwakili dengan nilai displasmen pada elemen *intersection line*.

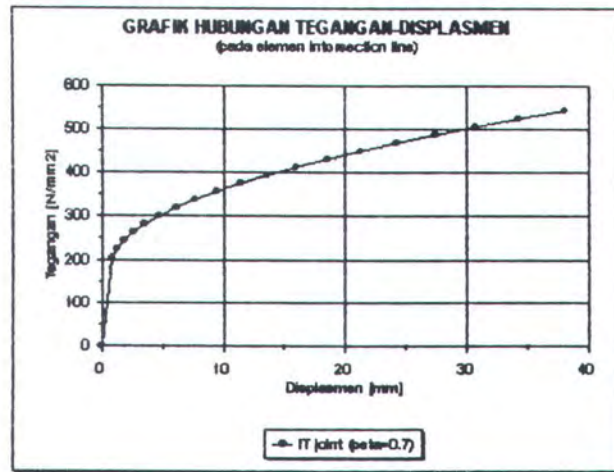




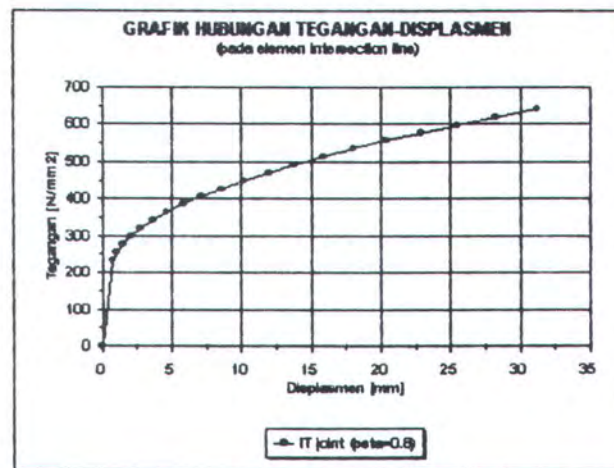
Gambar 6.7. Grafik Hubungan Tegangan-Displasmen (IT joint,  $\beta=0.5$ )



Gambar 6.8. Grafik Hubungan Tegangan-Displasmen (IT joint,  $\beta=0.6$ )

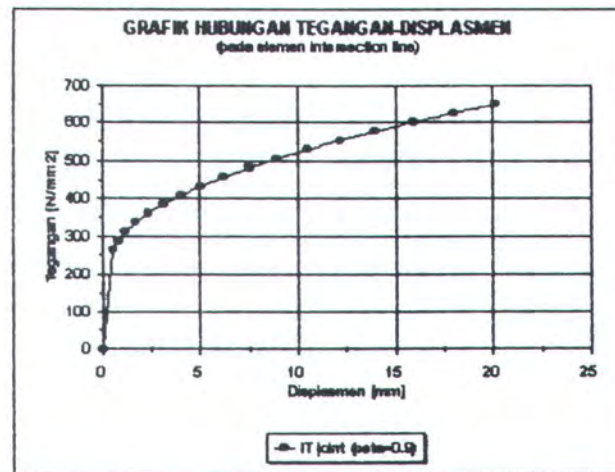


Gambar 6.9. Grafik Hubungan Tegangan-Displasmen (IT joint ,  $\beta=0.7$ )

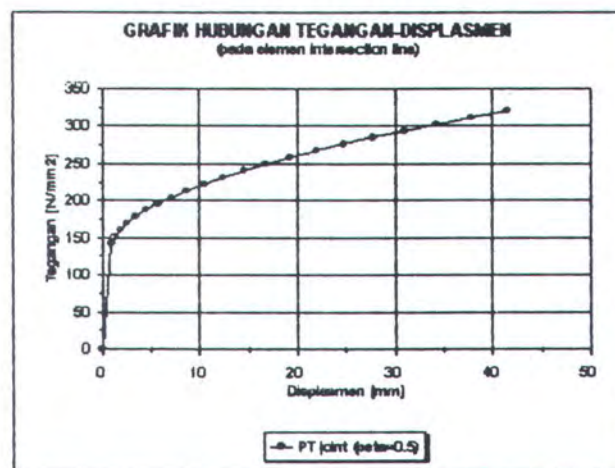


Gambar 6.10. Grafik Hubungan Tegangan-Displasmen (IT joint ,  $\beta=0.8$ )

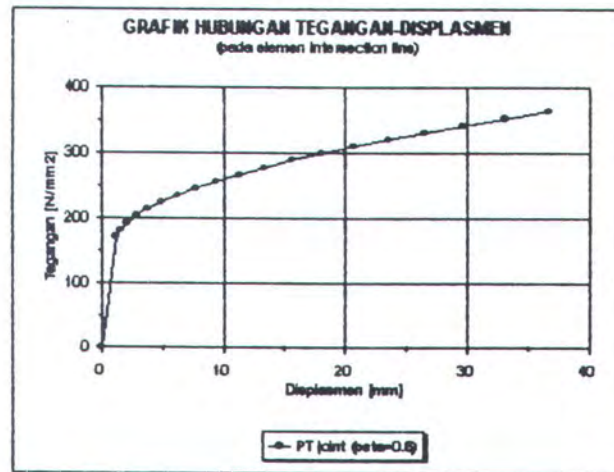




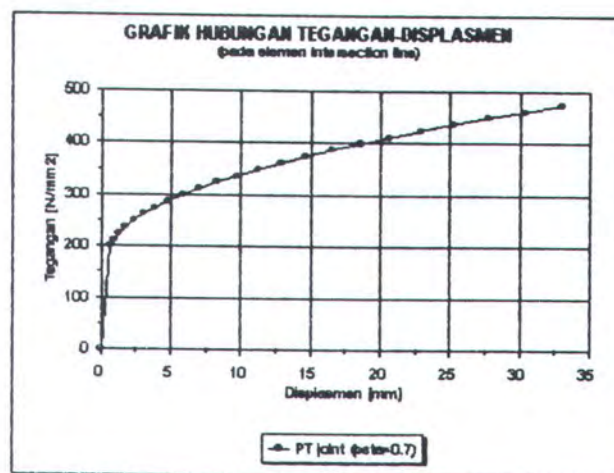
Gambar 6.11. Grafik Hubungan Tegangan-Displasmen (IT joint,  $\beta=0.9$ )



Gambar 6.12. Grafik Hubungan Tegangan-Displasmen (PT joint,  $\beta=0.5$ )

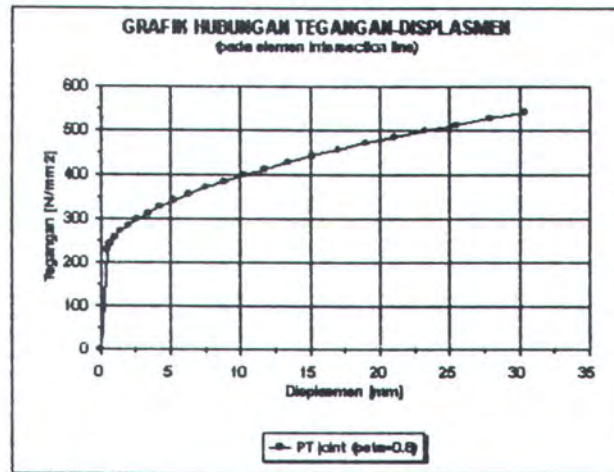


Gambar 6.13. Grafik Hubungan Tegangan-Displasmen (PT joint,  $\beta=0.6$ )

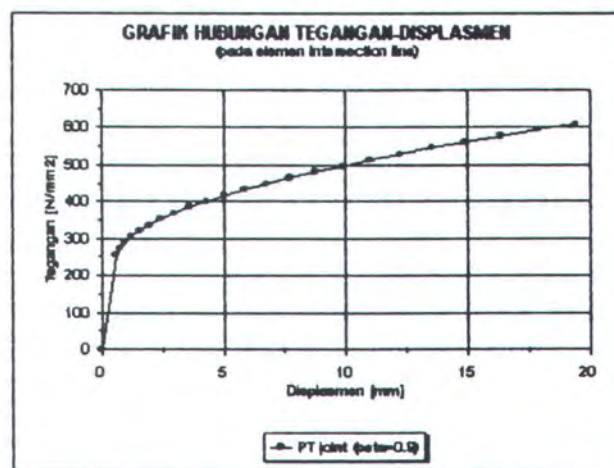


Gambar 6.14. Grafik Hubungan Tegangan-Displasmen (PT joint,  $\beta=0.7$ )





Gambar 6.15. Grafik Hubungan Tegangan-Displasmen (PT joint ,  $\beta=0.8$ )



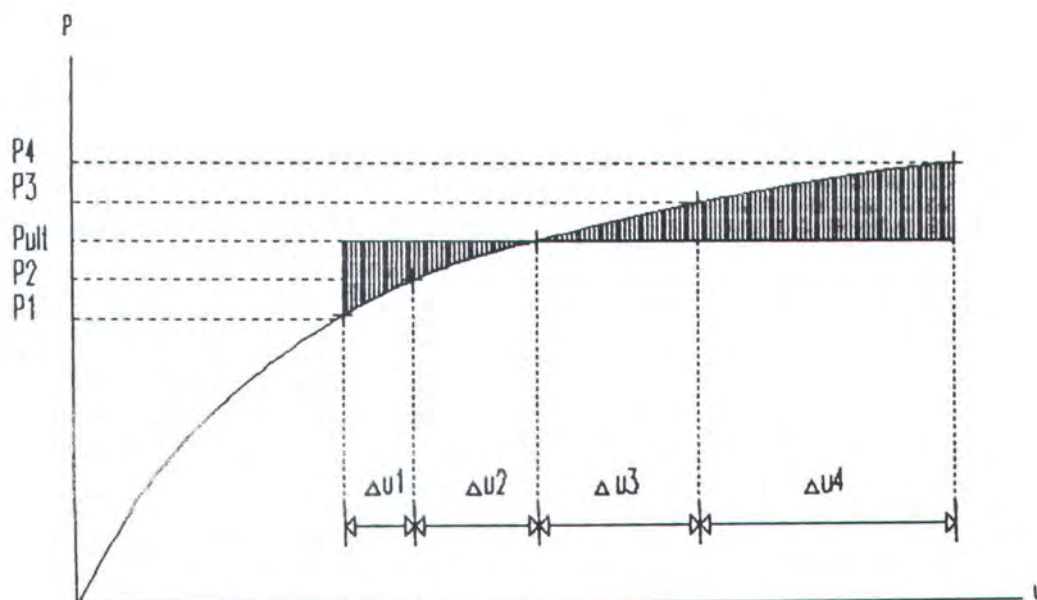
Gambar 6.16. Grafik Hubungan Tegangan-Displasmen (PT joint ,  $\beta=0.9$ )

Interpretasi pada gambar-gambar grafik di atas tentang hubungan antara tegangan dan deformasi adalah pada bagian-bagian awal proses incremental pembebanan, tegangan pada saat itu hanya menyebabkan perubahan yang kecil pada deformasi tetapi pada bagian-bagian pertengahan dan akhir dari proses incremental pembebanan maka tegangan pada saat itu akan menyebabkan perubahan yang relatif

lebih besar pada deformasi dibanding pada saat awal-awal pembebanan. Hal tersebut disebabkan oleh pengaruh plastisitas pada elemen ketika tegangan elemen tersebut melampaui dari tegangan yield material elemen.

Pengaruh plastisitas yang paling berarti adalah mempengaruhi nilai matrik kekakuan elemen struktur yang semakin kecil ketika elemen tersebut dalam kondisi plastis, hal ini dapat dihubungkan dengan persamaan (4.4-10) dan persamaan (4.4-11) yang dijelaskan sebelumnya. Kondisi ini selain menyebabkan perubahan nilai displasmen yang semakin besar juga menyebabkan penyelesaian konvergensi menjadi lebih sulit karena beban ekuivalen internal semakin besar untuk mengimbangi dari total beban eksternal yang ada, hal ini dapat dijelaskan pada hubungan persamaan (4.5-10) dimana tegangan yang terjadi juga semakin besar, hal ini seperti yang terlihat pada hubungan persamaan (4.5-1) dan persamaan (4.5-4).

Penyelesaian konvergensi sebelum dan sesudah  $P_{ultimate}$  dengan metode *tangent stiffness* seperti yang telah dijelaskan pada bab IV, dapat diinterpretasikan dengan gambar grafik berikut ini :



Gambar 6.17. Penyelesaian konvergensi sebelum dan sesudah  $P_{ultimate}$



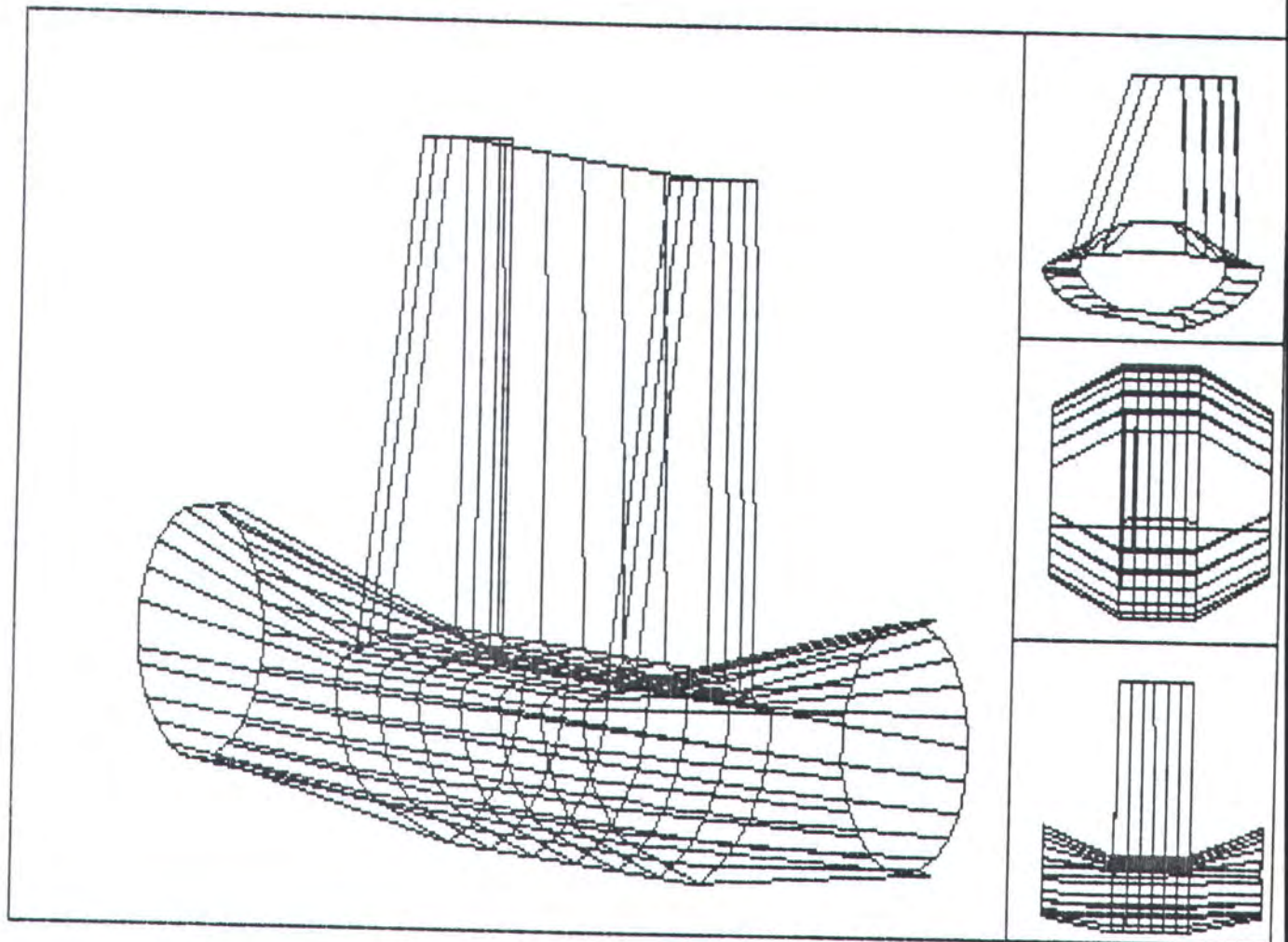
pada interpretasi gambar grafik di atas, penyelesaian konvergensi sebelum  $P_{ultimate}$  didekati dengan beban ekuivalen internal  $P_1$  dan  $P_2$  sebagai berikut :

- Proses keseimbangan gaya yang terjadi pada gambar grafik di atas oleh beban ekuivalen internal sebelum  $P_{ultimate}$  adalah  $(P_{ult}-P_1)$  dan  $(P_{ult}-P_2)$ .
- Proses keseimbangan gaya tersebut terus akan berjalan dengan perubahan nilai beban ekuivalen internal yang baru, hingga penyelesaian konvergensi tercapai.
- Pada grafik terlihat bahwa penyelesaian konvergensi akan tercapai karena perubahan nilai beban ekuivalen internal semakin mendekati  $P_{ultimate}$ .

sedangkan penyelesaian konvergensi sesudah  $P_{ultimate}$  didekati dengan beban ekuivalen internal  $P_3$  dan  $P_4$  sebagai berikut :

- Proses keseimbangan gaya yang terjadi adalah  $(P_{ult}-P_3)$  dan  $(P_{ult}-P_4)$ .
- Proses keseimbangan gaya terus berjalan dengan perubahan nilai beban ekuivalen yang semakin besar.
- Pada grafik terlihat bahwa penyelesaian konvergensi akan sulit tercapai karena nilai beban ekuivalen internal yang baru semakin menjauhi  $P_{ultimate}$ .

Bentuk deformasi yang terbesar akibat proses pembebanan tekan hingga beban mencapai *ultimate* adalah terjadi pada bagian tubular (*chord*) dengan arah sejajar pembebanan (arah sb. z) dan arah lateral pada sumbu tubular (arah sb. x) atau dalam hal ini deformasi tersebut menyebabkan perubahan bentuk oval pada tubular dan hal ini dapat dianggap sebagai moda kegagalan daripada model struktur yaitu dengan adanya perubahan deformasi yang relatif besar pada tubular (*chord*). Hal tersebut dapat digambarkan pada contoh gambar berikut yaitu mengenai bentuk deformasi model struktur setelah beban mencapai *ultimate*.



Gambar 6.18. Deformasi Tubular Setelah Pult





# **TUGAS AKHIR**

## **BAB VII**

### **KESIMPULAN DAN SARAN**



## BAB VII

### KESIMPULAN DAN SARAN

Dari pembahasan pada bab-bab terdahulu dapat diambil beberapa kesimpulan sebagai berikut :

1. Analisa secara numerik yang ideal untuk permasalahan kapasitas kekuatan *ultimate* dari sambungan adalah dengan pendekatan analisa statis non-linear pada lingkup permasalahan metode elemen hingga.
2. Proses non-linear yang terjadi adalah berdasarkan pada ketidaklinearan material (*material nonlinearity*) akibat perubahan kondisi elastis-plastis, sedangkan pendekatan nilai kapasitas kekuatan *ultimate* sambungan berdasarkan pada ketidakseimbangan antara beban eksternal dengan beban ekuivalen internal yang terjadi atau proses *un*-konvergensi dimana penyelesaian konvergensi sulit dicapai karena pengaruh plastisitas yang terjadi pada tegangan elemen.
3. Hasil verifikasi atau perbandingan kapasitas kekuatan *ultimate* sambungan IT dan PT *joint* antara hasil analisa non-linear dan hasil penurunan rumus terhadap hasil pengujian menunjukkan bahwa hasil analisa non-linear dan hasil penurunan rumus Kurobane memberikan hasil yang lebih baik daripada hasil penurunan rumus Wardenier dan Kamba.
4. Pengaruh perubahan parameter  $\beta$  terhadap kapasitas kekuatan *ultimate* adalah konsisten secara non-linear.

Saran dan pengembangan lebih lanjut yang dapat diberikan dari pembahasan permasalahan dalam tugas akhir ini adalah sebagai berikut :



- Mengefisiensikan program penyelesaian permasalahan non-linear hingga kapasitas program dapat digunakan untuk model elemen yang lebih baik.
- Mengembangkan permasalahan kapasitas kekuatan *ultimate* sambungan IT dan PT *joint* untuk kondisi pembebanan yang berbeda seperti beban aksial tarik dan beban momen (*in plane bending* atau *out plane bending*) atau kombinasi beban aksial dan beban momen.
- Meninjau pengaruh parameter geometri sambungan yang lain terhadap kapasitas kekuatan *ultimate* sambungan IT dan PT *joint*, seperti  $\gamma$  ( $D/2T$ ) dan  $\eta$  ( $h/D$ ).



# **TUGAS AKHIR**

**DAFTAR PUSTAKA**



## DAFTAR PUSTAKA

1. R. L. STEINMETZ, *Ultimate Capacity of I-beam to Tubular (IT) Joints*, Compiled by Mc Demmort's paper, June, 1986.
2. D. R. J. OWEN, E. HINTON, *Finite Element Software for Plates and Shells*, Pineridge Press, Swansea, U. K., 1984.
3. ROBERT D. COOK, DAVID S. MALKUS, MICHAEL E. PLESHA, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons Inc., Canada, 3rd edition, 1989.
4. K. J. BATHE, *Finite Element Procedures in Engineering Analysis*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1982.
5. AMERICAN PETROLEUM INSTITUTE (API), *Recommended Practice for Planning, Designing, and Constructing Fixed Offshore Platform*, API RP2A, 16th edition, 1986.
6. J. A. YURA, *Ultimate Capacity Equations for Tubular Joints*, Offshore Technology Conference (OTC), Houston, Texas, May, 1980.
7. Ir. ASYHAR IMRON, Msc., *Kumpulan Catatan Mata Kuliah Metode Elemen Hingga*, Fakultas Teknologi Kelautan, ITS.





# **TUGAS AKHIR**

## **LAMPIRAN 1**

### **LISTING PROGRAM UTAMA**



## LAMPIRAN 1. LISTING PROGRAM UTAMA

```
C234567
C LISTING PROGRAM ANALISA ELASTO-PLASTIC UNTUK PLATE DAN SHELL
C
C$NOWARN
PROGRAM PLSHELL
C
  DIMENSION ASDIS(2080),COORD(416,8),ELOAD(132,45),EQRHS(10),
.      EQUAT(370,10),EFFST(1188),EPSTN(1188),ESTIF(45,45),
.      FIXED(2080),GLOAD(370),GSTIF(2850),GRAVI(3),
.      IFFIX(2080),LOCEL(45),LNODS(132,9),MATNO(132,10),
.      NACVA(370),NAMEV(10),NDEST(45),NDFRO(132),
.      NOFIX(32),NOUTP(2),NPIVO(10),POSGP(5),THICK(416),
.      PRESC(32,5),PROPS(5,17),RLOAD(132,45),STFOR(2080),
.      STRSG(5,1188),TDISP(2080),TLOAD(132,45),TOFOR(2080),
.      TREAC(32,5),VECRV(370),WEIGP(5),DICOS(3,1248),
.      AMATX(9,5),DMATT(5,5,5)
C
C SET VARIABEL DATA
C
  CALL DIMEN(MBUFA,MELEM,MEVAB,MFRON,MMATS,MPOIN,MSTIF,
.      MTOTG,MTOTV,MVFIX,NDOFN,NPROP,NSTRE,M3POI,
.      MLAYR)
C
C BACA DATA INPUT
C
  CALL INPUT(ANVEL,COORD,GRAVI,IFFIX,LNODS,
.      MATNO,MFRON,MELEM,MMATS,MPOIN,
.      MTOTV,MVFIX,NDFRO,NDOFN,NELEM,NCOLA,
.      NEVAB,NGAUS,NGAUZ,NMATS,NNODE,MLAYR,
.      NOFIX,NPOIN,NPROP,NTOTG,NLAYR,NREST,
.      NTOTV,NVFIX,POSGP,PRESC,PROPS,WEIGP,
.      NALGO,NINCS,LARGE)
C
C HITUNG MATRIK ELASTISITAS [D] DAN MATRIK PARAMETER ANISOTROPIC
C
  CALL MODAN(AMATX,DMATT,NMATS,NPROP,PROPS,MMATS,
.      MATNO,MELEM,MLAYR,NELEM,NLAYR)
C
C SET SISTEM KOORDINAT ORTHOGONAL UNTUK TIAP NODAL POINT
C
  CALL WORKS(COORD,DICOS,LNODS,THICK,MELEM,MPOIN,
.      NPOIN,M3POI)
C
C HITUNG MATRIK B DAN MATRIK G
C
  CALL BGMAT(COORD,DICOS,LNODS,MATNO,MELEM,
.      MLAYR,MMATS,MPOIN,M3POI,NELEM,
.      NEVAB,NGAUS,NGAUZ,NLAYR,NNODE,NPROP,
.      POSGP,PROPS,THICK,WEIGP)
C
C HITUNG NODAL FORCE AKIBAT BEBAN LUAR YANG BEKERJA
C
  CALL LOADS(ANVEL,COORD,RLOAD,GRAVI,LNODS,
.      MATNO,MELEM,MEVAB,MMATS,MPOIN,DICOS,
.      NELEM,NEVAB,NGAUS,THICK,
.      NNODE,NPROP,NSTRE,POSGP,M3POI,
.      PROPS,WEIGP,MLAYR,NLAYR)
C
C INITIALISE CERTAIN ARRAYS
C
  CALL ZERO(EFFST,ELOAD,EPSTN,MELEM,MEVAB,KINCS,
.      MTOTG,MTOTV,NDOFN,NELEM,NEVAB,NREST,
```

```

      NSTRE, NTOTG, NTOTV, NVFIX, MVFIX, STRSG,
      TDISP, TFACT, TLOAD, TREAC)
C
C LOOP OVER EACH INCREMENT
C
      DO 100 IINCS=1, NINCS
C
C BACA HASIL DATA DARI TIAP INCREMENT
C
      CALL      INCREM(ELOAD, FIXED, IINCS, MELEM, MEVAB, MITER,
      .          MTOTV, MVFIX, NDOFN, NELEM, NEVAB, NOUTP,
      .          NOFIX, NTOTV, NVFIX, PRESC, RLOAD, TFACT,
      .          TLOAD, TOLER, LNODS, IFFIX, NNODE, NCOLA,
      .          NREST, KINCS)
C
C LOOP OVER EACH ITERATION
C
      KSTOP=0
      KUNLO=0
      DO 50 IITER=1, MITER
      KITER=IINCS+IITER
      JINCS=IINCS-KINCS
C
C SET INDICATOR UNTUK ALGORITMA PENYELESAIAN KEKAKUAN ELEMEN
C
      CALL      ALGOR(FIXED, KITER, IITER, KRESL, MTOTV, NALGO,
      .          NTOTV, KUNLO, KINCS)
C
C PERIKSA MATRIK KEKAKUAN
C
      IF (KRESL.EQ.1)
      .CALL      STIFF (EPSTN, ESTIF, KITER, LNODS, MATNO,
      .          MELEM, MEVAB, MMATS, MPOIN, MTOTG, NDOFN,
      .          NELEM, NEVAB, NGAUS, NNODE, NPROP,
      .          NSTRE, POSGP, PROPS, STRSG, WEIGP, AMATX,
      .          DMATT, MLAYR, NLAYR, THICK,
      .          TDISP, MTOTV, LARGE)
C
C PENYELESAIAN PERSAMAAN SIMULTAN DENGAN METODE FRONTAL
C
      CALL      FRONT (ASDIS, ELOAD, BQRHS, EQUAT, ESTIF, FIXED,
      .          GLOAD, GSTIF, IFFIX, JINCS, IITER, KRESL,
      .          LOCEL, LNODS, MBUFA, MELEM, MEVAB, MFRON,
      .          MSTIF, MTOTV, MVFIX, NACVA, NAMEV, NDEST,
      .          NDOFN, NELEM, NEVAB, NNODE, NOFIX, NPIVO,
      .          NPOIN, NTOTV, TDISP, TLOAD, TREAC, VECRV)
C
C HITUNG RESIDUAL FORCE
C
      CALL      RESTR (ASDIS, EFFST, ELOAD, LNODS,
      .          MATNO, MELEM, MMATS, MPOIN, MTOTG, MTOTV,
      .          NDOFN, NELEM, NEVAB, NGAUS, NNODE,
      .          NPROP, NSTRE, POSGP, PROPS, STRSG,
      .          TDISP, WEIGP, EPSTN, KUNLO, AMATX, DMATT,
      .          THICK, MLAYR, NLAYR, LARGE)
C
C CHECK FOR CONVERGENCE
C
      CALL      CONVER (ELOAD, IITER, LNODS, MELEM, MEVAB, MTOTV,
      .          NCHEK, NDOFN, NELEM, NEVAB, NNODE, NTOTV,
      .          STFOR, TLOAD, TOFOR, TOLER)
C
C CETAK HASIL OUTPUT PROGRAM
C
      IF (IITER.EQ.1.AND.NOUTP(1).GT.0)

```



```

      .CALL      OUTPUT(IITER,MTOTG,MTOTV,MVFIX,NCHEK,NELEM,
      .          NGAUS,NOFIX,NOUTP,NPOIN,
      .          NSTRE,NVFIX,STRSG,TDISP,TREAC,EPSTN,
      .          POSGP,EFFST,MATNO,MMATS,PROPS,
      .          NPROP,MELEM,THICK,
      .          MPOIN,LNODS,MLAYR,NLAYR)
C
C JIKA HASIL PENYELESAIAN SUDAH CONVERGENCE DAN CETAK HASIL OUTPUT
C
      IF(IITER.EQ.1.AND.NCHEK.EQ.0) GO TO 100
      IF(NCHEK.EQ.0) GO TO 75
50  CONTINUE
C
C
      KSTOP=1
75  CALL      OUTPUT(IITER,MTOTG,MTOTV,MVFIX,NCHEK,NELEM,
      .          NGAUS,NOFIX,NOUTP,NPOIN,
      .          NSTRE,NVFIX,STRSG,TDISP,TREAC,EPSTN,
      .          POSGP,EFFST,MATNO,MMATS,PROPS,
      .          NPROP,MELEM,THICK,
      .          MPOIN,LNODS,MLAYR,NLAYR)
      IF(KSTOP.EQ.1) STOP
C
C SIMPAN HASIL OUTPUT SEBAGAI DATA UNTUK PROSES INCREMENT SELANJUTNYA
C
      CALL      RESTAR(EFFST,ELOAD,EPSTN,MELEM,MEVAB,MTOTG,
      .          MTOTV,MVFIX,TDISP,TLOAD,TREAC,STRSG,
      .          TFACT,IINCS)
100 CONTINUE
      STOP
      END

      SUBROUTINE DIMEN(MBUFA,MELEM,MEVAB,MFRON,MMATS,MPOIN,MSTIF,MTOTG,
      .          MTOTV,MVFIX,NDOFN,NPROP,NSTRE,M3POI,MLAYR)
C
C SUBROUTINE UNTUK SET VARIABEL DATA
C
      MBUFA=10
      MELEM=132
      MFRON=370
      MLAYR=1
      MMATS=5
      MPOIN=416
      M3POI=3*MPOIN
      NDOFN=5
      NSTRE=5
      MEVAB=NDOFN*9
      MSTIF=(MFRON+1)*MFRON/2
      MTOTG=MELEM*MLAYR*9
      MTOTV=MPOIN*NDOFN
      MVFIX=36
      NPROP=17
      RETURN
      END

      SUBROUTINE INPUT(ANVEL,COORD,GRAVI,IFFIX,LNODS,
      .          MATNO,MFRON,MELEM,MMATS,MPOIN,
      .          MTOTV,MVFIX,NDFRO,NDOFN,NELEM,NCOLA,
      .          NEVAB,NGAUS,NGAUZ,MMATS,NNODE,MLAYR,
      .          NOFIX,NPOIN,NPROP,NTOTG,NLAYR,NREST,
      .          NTOTV,NVFIX,POSGP,PRESC,PROPS,WEIGP,
      .          NALGO,NINCS,LARGE)
C
C SUBROUTINE UNTUK MEMBACA DATA INPUT
C

```

```

        DIMENSION COORD(MPOIN,8), IFFIX(MTOTV), LNODS(MELEM,9),
        .          MATNO(MELEM,MLAYR), NDFRO(MELEM), GRAVI(3),
        .          NOFIX(MVFIX), POSGP(5), PRESC(MVFIX,NDOFN),
        .          PROPS(MMATS,NPROP), TITLE(12), WEIGP(5)
        READ(5,920) TITLE
        WRITE(6,920) TITLE
920 FORMAT(12A6)
C
C MEMBACA DATA PERTAMA
C
        READ(101,900) NPOIN, NELEM, NVFIX, NNODE, NMATS, NGAUS, NGAUZ, NCOLA,
        . NALGO, NINCS, NLAYR, LARGE, NREST
900 FORMAT(16I5, /5X, 15I5)
        NEVAB=NDOFN*NNODE
        NTOTV=NPOIN*NDOFN
        NGAUZ=NGAUS*NGAUS
        NTOTG=NELEM*NGAUZ*NLAYR
        WRITE(6,901) NPOIN, NELEM, NVFIX, NNODE, NMATS, NGAUS, NGAUZ,
        . NEVAB, NCOLA, NALGO, NINCS, NLAYR, LARGE, NREST
901 FORMAT(///, 5X, 8H NPOIN =, I5/5X, 8H NELEM =, I5/5X, 8H NVFIX =, I5/
        . 5X, 8H NNODE =, I5/5X, 8H NMATS =, I5/5X, 8H NGAUS =, I5/
        . 5X, 8H NGAUZ =, I5/5X, 8H NEVAB =, I5/5X, 8H NCOLA =, I5/5X,
        . 8H NALGO =, I5/5X, 8H NINCS =, I5, /,
        . 5X, 8H NLAYR =, I5/5X, 8H LARGE =, I5/5X, 8H NREST =, I5)
        WRITE(6,912)
        READ(102,913) GRAVI(1), GRAVI(2), GRAVI(3), ANVEL
        WRITE(6,913) GRAVI(1), GRAVI(2), GRAVI(3), ANVEL
912 FORMAT(//41H X-GRAVITY Y-GRAVITY Z-GRAVITY ANG VEL/)
913 FORMAT(4F10.5)
        CALL CHECK1(NDOFN, NELEM, NGAUS, NMATS, NNODE, NPOIN,
        . MMATS, NVFIX, NGAUZ, NLAYR)
C
C BACA HUBUNGAN ELEMEN NODAL POIN DAN ELEMEN PROPERTY
C
        WRITE(6,902)
902 FORMAT(//8H ELEMENT, 5X, 15H PROPERTY/LAYER, 35X, 12H NODE NUMBERS)
        DO 2 ILEM=1, NELEM
        READ(103,900) NUMEL, (MATNO(NUMEL, ILAYR), ILAYR=1, NLAYR),
        . (LNODS(NUMEL, INODE), INODE=1, NNODE)
        WRITE(6,903) NUMEL, (MATNO(NUMEL, ILAYR), ILAYR=1, NLAYR)
        2 WRITE(6,940) (LNODS(NUMEL, INODE), INODE=1, NNODE)
903 FORMAT(1X, I5, 4X, 10I5)
940 FORMAT(60X, 12I5)
C
C ZERO ALL THE NODAL COORDINATES
C
        DO 4 IPOIN=1, NPOIN
        DO 4 IDIME=1, 8
        4 COORD(IPOIN, IDIME)=0.0
C
C BACA DATA NODAL KOORDINAT
C
        WRITE(6,904)
904 FORMAT(//5H NODE, 8X, 1HZ, 14X, 1HY, 14X, 1HZ, 13X, 5HPRESS)
        5 READ(104,905) IPOIN, (COORD(IPOIN, IDIME), IDIME=1, 8)
905 FORMAT(I5, 4F15.10/5X, 4F15.10)
        IF(IPOIN.NE.NPOIN) GO TO 6
C
C INTERPOLATE COORDINATES OF MID-SIDE NODES
C
        CALL NODEX(COORD, LNODS, MELEM, MPOIN, NELEM, NNODE)
        DO 10 IPOIN=1, NPOIN
        10 WRITE(6,906) IPOIN, (COORD(IPOIN, IDIME), IDIME=1, 8)
906 FORMAT(I5, 4F15.10/5X, 4F15.10)
C

```



C BACA DATA DEDREES OF FREEDOM

C

```

      WRITE(6,907)
907  FORMAT(/5H NODE,6X,4HCODE,15X,12HFIXED VALUES)
      DO 8 IVFIX=1,NVFIX
        READ(105,908) NOFIX(IVFIX),IFPRE,(PRESC(IVFIX,IDOFN),IDOFN=1,
        .NDOFN)
        WRITE(6,908) NOFIX(IVFIX),IFPRE,(PRESC(IVFIX,IDOFN),IDOFN=1,NDOFN)
        NLOCA=(NOFIX(IVFIX)-1)*NDOFN
        IFDOF=10** (NDOFN-1)
        DO 8 IDOFN=1,NDOFN
          NGASH=NLOCA+IDOFN
          IF(IFPRE.LT.IFDOF) GO TO 8
          IFFIX(NGASH)=1
          IFPRE=IFPRE-IFDOF
        8 IFDOF=IFDOF/10
908  FORMAT(I5,5X,I5,5F10.6)

```

C

C BACA PENENTUAN ELEMEN PROPERTIES

C

```

      WRITE(6,910)
910  FORMAT(/7H  NUMAT,10X,18HELEMENT PROPERTIES)
      DO 18 IMATS=1,NMATS
        READ(106,900) NUMAT
        READ(107,930) (PROPS(NUMAT,IPROP),IPROP=1,NPROP)
930  FORMAT(7F10.5/7F10.5/7F10.5)
      18 WRITE(6,911) NUMAT,(PROPS(NUMAT,IPROP),IPROP=1,NPROP)
911  FORMAT(1X,I4,3X,7E15.5/,8X,7E15.5/,8X,7E15.5)

```

C

C SET GAUSSIAN INTEGRATION CONSTANTS

C

```

      CALL      GAUSSQ(NGAUS,POSGP,WEIGP)
      CALL      CHECK2(COORD,IFFIX,LNODS,MATNO,MELEM,MFRON,MPOIN,MTOTV,
      .          MVFIX,NDFRO,NDOFN,NELEM,NMATS,NNODE,NOFIX,NPOIN,
      .          NVFIX,NLAYR)
      RETURN
      END

```

```

      SUBROUTINE CHECK1(NDOFN,NELEM,NGAUS,NMATS,NNODE,NPOIN,
      .          MMATS,NVFIX,NGAUZ,NLAYR)

```

C

C SUBROUTINE UNTUK MEMERIKSA DATA INPUT

C

```

      DIMENSION NEROR(20)
      DO 10 IEROR=1,8
10  NEROR(IEROR)=0

```

C

C CREATE THE DIAGNOSTIC MESSAGES

C

```

      IF(NPOIN.LE.0) NEROR(1)=1
      IF(NELEM*NNODE.LT.NPOIN) NEROR(2)=1
      IF(NVFIX.LT.2.OR.NVFIX.GT.NPOIN) NEROR(3)=1
      IF(NNODE.LT.8.OR.NNODE.GT.9) NEROR(4)=1
      IF(NDOFN.NE.5.OR.NLAYR.GT.10) NEROR(5)=1
      IF(NMATS.LT.1.OR.NMATS.GT.MMATS) NEROR(6)=1
      IF(NGAUS.LT.2.OR.NGAUS.GT.3) NEROR(7)=1
      IF(NGAUZ.LT.2.OR.NGAUZ.GT.3) NEROR(8)=1

```

C

C PRINT THE ERRORS DIAGNOSED

C

```

      KEROR=0
      DO 20 IEROR=1,8
      IF(NEROR(IEROR).EQ.0) GO TO 20
      KEROR=1
      WRITE(6,900) IEROR

```

```

900 FORMAT(/31H *** DIAGNOSIS BY CHECK1, ERROR,I3)
20 CONTINUE
   IF (KEROR.EQ.0) RETURN
C
C ECHO ALL THE REMAINING DATA WITHOUT FURTHER COMMENT
C
   CALL ECHO
   END

   SUBROUTINE CHECK2 (COORD, IFFIX, LNODS, MATNO, MELEM, MFRON, MPOIN, MTOTV,
   .                 MVFIX, NDFRO, NDOFN, NELEM, NMATS, NNODE, NOFIX, NPOIN,
   .                 NVFIX, NLAYR)
C
C SUBROUTINE UNTUK MEMERIKSA DATA INPUT
C
   DIMENSION COORD (MPOIN,8), IFFIX (MTOTV), LNODS (MELEM,12),
   .         MATNO (MELEM, NLAYR), NDFRO (MELEM), NEROR (20), NOFIX (MVFIX)
C
C CEK KOORDINAT NODAL POINT
C
   DO 5 IEROR=9,20
   5 NEROR (IEROR)=0
   DO 10 IELEM=1, NELEM
   10 NDFRO (IELEM)=0
   DO 40 IPOIN=2, NPOIN
   KPOIN=IPOIN-1
   DO 30 JPOIN=1, KPOIN
   DO 20 IDIME=1,3
   IF (COORD (IPOIN, IDIME) .NE. COORD (JPOIN, IDIME)) GO TO 30
   20 CONTINUE
   NEROR (9)=NEROR (9)+1
   30 CONTINUE
   40 CONTINUE
C
C CEK JUMLAH ELEMEN PROPERTI
C
   DO 50 IELEM=1, NELEM
   DO 50 ILAYR=1, NLAYR
   50 IF (MATNO (IELEM, ILAYR) .GT. NMATS) NEROR (10)=NEROR (10)
   . +1
C
C CEK PENOMORAN NODE PADA ELEMEN
C
   DO 70 IELEM=1, NELEM
   DO 60 INODE=1, NNODE
   IF (LNODS (IELEM, INODE) .EQ.0) NEROR (11)=NEROR (11)+1
   60 IF (LNODS (IELEM, INODE) .LT.0. OR. LNODS (IELEM, INODE) .GT. NPOIN) NEROR (
   . 12)=NEROR (12)+1
   70 CONTINUE
C
C CEK PERULANGAN ELEMEN NODE NUMBER
C
   DO 140 IPOIN=1, NPOIN
   KSTAR=0
   DO 100 IELEM=1, NELEM
   KZERO=0
   DO 90 INODE=1, NNODE
   IF (LNODS (IELEM, INODE) .NE. IPOIN) GO TO 90
   KZERO=KZERO+1
   IF (KZERO.GT.1) NEROR (13)=NEROR (13)+1
C
   IF (KSTAR.NE.0) GO TO 80
   KSTAR=IELEM
C
   NDFRO (IELEM)=NDFRO (IELEM)+NDOFN

```



```

      80 CONTINUE
C
      KLAST=IELEM
      NLAST=INODE
      90 CONTINUE
      100 CONTINUE
         IF(KSTAR.EQ.0) GO TO 110
         IF(KLAST.LT.NELEM) NDFRO(KLAST+1)=NDFRO(KLAST+1)-NDOFN
         LNODS(KLAST,NLAST)=-IPOIN
         GO TO 140
C
C CHECK UNUSED NODE NUMBER
C
      110 WRITE(6,900) IPOIN
      900 FORMAT(/15H CHECK WHY NODE,I4,14H NEVER APPEARS)
         NEROR(14)=NEROR(14)+1
         SIGMA=0.0
         DO 120 IDIME=1,3
      120 SIGMA=SIGMA+ABS(COORD(IPOIN,IDIME))
         IF(SIGMA.NE.0.0) NEROR(15)=NEROR(15)+1
C
C CHECK THAT UNUSED NODE NUMBER IS NOT A RESTRAINED NODE
C
         DO 130 IVFIX=1,NVFIX
      130 IF(NOFIX(IVFIX).EQ.IPOIN) NEROR(16)=NEROR(16)+1
      140 CONTINUE
C
C CALCULATE THE LARGEST FRONTWIDTH
C
         NFRON=0
         KFRON=0
         DO 150 IELEM=1,NELEM
            NFRON=NFRON+NDFRO(IELEM)
      150 IF(NFRON.GT.KFRON) KFRON=NFRON
         WRITE(6,905) KFRON
      905 FORMAT(/33H MAXIMUM FRONTWIDTH ENCOUNTERED =,I5/)
         IF(KFRON.GT.MFRON) NEROR(17)=1
C
C CHECK THE FIXED VALUES
C
         DO 170 IVFIX=1,NVFIX
            IF(NOFIX(IVFIX).LE.0.OR.NOFIX(IVFIX).GT.NPOIN) NEROR(18)=NEROR(18)
            +1
            KOUNT=0
            NLOCA=(NOFIX(IVFIX)-1)*NDOFN
            DO 160 IDOFN=1,NDOFN
               NLOCA=NLOCA+1
      160 IF(IVFIX(NLOCA).GT.0) KOUNT=1
            IF(KOUNT.EQ.0) NEROR(19)=NEROR(19)+1
            KVFIX=IVFIX-1
            DO 170 JVFIX=1,KVFIX
      170 IF(IVFIX.NE.1.AND.NOFIX(IVFIX).EQ.NOFIX(JVFIX)) NEROR(20)=NEROR(20)
            +1
            KEROR=0
            DO 180 IEROR=9,20
               IF(NEROR(18).EQ.0) GO TO 180
               KEROR=1
               WRITE(6,910) IEROR,NEROR(18)
      910 FORMAT(/31H *** DIAGNOSIS BY CHECK2, ERROR,I3,6X,18H ASSOCIATED N
            UMBER,I5)
      180 CONTINUE
         IF(KEROR.NE.0) GO TO 200
C
C RETURN ALL NODAL CONNECTION NUMBERS TO POSITIVE VALUES
C

```

```

      DO 190 IELEM=1,NELEM
      DO 190 INODE=1,NNODE
190  LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
200  CALL ECHO
      END

      SUBROUTINE ECHO
C
C  JIKA TERDAPAT INPUT DATA YANG ERROR MAKA SUBROUTINE INI AKAN
C  TETAP MENYIMPAN INPUT DATA YANG ADA
C
      DIMENSION NTITL(80)
      WRITE(6,900)
900  FORMAT(/ /50H NOW FOLLOWS A LISTING OF POST-DISASTER DATA CARDS/)
      10  READ(5,905) NTITL
905  FORMAT(80A1)
      WRITE(6,910) NTITL
910  FORMAT(20X,80A1)
      GO TO 10
      END

      SUBROUTINE NODEX(COORD, LNODS, MELEM, MPOIN, NELEM, NNODE)
C
C  SUBROUTINE UNTUK MEMERIKSA MID-SIDE NODES
C
      DIMENSION COORD(MPOIN,8), LNODS(MELEM,9), ELCOR(8,8)
C
C  LOOP TIAP ELEMEN
C
      DO 60 IELEM=1,NELEM
C
C  LOOP OVER EACH ELEMENT EDGE
C
      NNOD1=7
      DO 20 INODE=1,NNOD1,2
C
C  HITUNG NODE PERTAMA
C
      NODST=LNODS(IELEM,INODE)
      IGASH=INODE+2
      IF(IGASH.GT.8) IGASH=1
C
C  HITUNG NODE TERAKHIR
C
      NODFN=LNODS(IELEM,IGASH)
      MIDPT=INODE+1
C
C  HITUNG INTERMEDIATE NODE
C
      NODMD=LNODS(IELEM,MIDPT)
      TOTAL=ABS(COORD(NODMD,1))+ABS(COORD(NODMD,2))+ABS(COORD(NODMD,3))
C
C  JIKA INTERMEDIATE NODE ADALAH NOL MAKA DILAKUKAN INTERPOLASI
C
      IF(TOTAL.GT.0.0) GO TO 20
      KOUNT=1
      10  COORD(NODMD,KOUNT)=(COORD(NODST,KOUNT)+COORD(NODFN,KOUNT))/2.0
      KOUNT=KOUNT+1
      IF(KOUNT.LE.8) GO TO 10
      20  CONTINUE
      IF(NNODE.EQ.8) GO TO 60
C
C  SET UP CENTRAL NODE UNTUK ELEMEN DENGAN 9 NODE
C

```



```

      NODCE=LNODS(IELEM,9)
      DO 30 INODE=1,8
      NODEB=LNODS(IELEM,INODE)
      DO 30 IDIME=1,8
30  ELCOR(IDIME,INODE)=COORD(NODEB,IDIME)
      DO 50 IDIME=1,8
      CENCO=0.0
      DO 35 INODE=1,7,2
35  CENCO=CENCO+ELCOR(IDIME,INODE)
      CENCO=CENCO*(-0.5)
      DO 40 INODE=2,8,2
40  CENCO=CENCO+ELCOR(IDIME,INODE)
      CENCO=CENCO*0.5
50  COORD(NODCE,IDIME)=CENCO
60  CONTINUE
      RETURN
      END

      SUBROUTINE GAUSSQ(NGAUS,POSGP,WEIGP)
C
C  SUBROUTINE UNTUK MENETAPKAN KONSTANTA GAUSS-LAGENDRE INTEGRATION
C
      DIMENSION POSGP(5), WEIGP(5)
      DO 2 IGASH=1,5
      POSGP(IGASH)=0.0
2  WEIGP(IGASH)=0.0
      IF(NGAUS.GT.2) GO TO 4
      POSGP(1)=-0.577350269189626
      WEIGP(1)=1.0
      GO TO 8
4  POSGP(1)=-0.774596669241483
      POSGP(2)=0.0
      WEIGP(1)=0.5555555555555556
      WEIGP(2)=0.8888888888888889
8  KGAUS=NGAUS/2
      DO 10 IGASH=1,KGAUS
      JGASH=NGAUS+1-IGASH
      POSGP(JGASH)=-POSGP(IGASH)
      WEIGP(JGASH)=WEIGP(IGASH)
10  CONTINUE
C
C  EXTRA KONSTANTA UNTUK SELECTIVE INTEGRATION
C
      POSGP(4)=-0.577350269189626
      WEIGP(4)=1.0
      POSGP(5)=-POSGP(4)
      WEIGP(5)=WEIGP(4)
      RETURN
      END

      SUBROUTINE MODAN(AMATX,DMATT,NMATS,NPROP,PROPS,MMATS,
      MATNO,MELEM,MLAYR,NELEM,NLAYR)
C
C  SUBROUTINE UNTUK MENGHITUNG MATRIK ELASTISITAS DAN MATRIK
C  PARAMETER ANISOTROPIK
C
      DIMENSION AMATX(9,MMATS),DMATT(5,5,MMATS),PROPS(MMATS,NPROP),
      APARA(5,5),TRANS(5,5),GASHM(5,5),MATNO(MELEM,MLAYR),
      COEFE(2)
      DO 15 IMATS=1,NMATS
C
C  SET MATRIK PARAMETER ANISOTROPIK
C
      UNIAX=PROPS(IMATS,6)
      DO 5 I=1,9

```

```

5  AMATX(I,IMATS)=0.0
   AMATX(1,IMATS)=1.0
   AMATX(4,IMATS)=(UNIAX/PROPS(IMATS,12))*2.0
   A3Z=(UNIAX/PROPS(IMATS,13))*2.0
   AMATX(6,IMATS)=(UNIAX/PROPS(IMATS,14))*2.0
   AMATX(2,IMATS)=2.0*A3Z-0.5*(1.0+AMATX(4,IMATS)+AMATX(6,IMATS))
   AMATX(7,IMATS)=(UNIAX/PROPS(IMATS,15))*2.0
   AMATX(9,IMATS)=(UNIAX/PROPS(IMATS,16))*2.0
C
C SET MATRIK ELASTISITAS
C
   GASH=1.0-PROPS(IMATS,2)**2.0*PROPS(IMATS,8)/PROPS(IMATS,1)
   DO 10 I=1,5
   DO 10 J=1,5
10  DMATT(I,J,IMATS)=0.0
   DMATT(1,1,IMATS)=PROPS(IMATS,1)/GASH
   DMATT(2,2,IMATS)=PROPS(IMATS,8)/GASH
   DMATT(1,2,IMATS)=PROPS(IMATS,2)*DMATT(2,2,IMATS)
   DMATT(2,1,IMATS)=DMATT(1,2,IMATS)
   DMATT(3,3,IMATS)=PROPS(IMATS,9)
   DMATT(4,4,IMATS)=PROPS(IMATS,10)
   DMATT(5,5,IMATS)=PROPS(IMATS,11)
15  CONTINUE
C
C HITUNG SHEAR CORRECTION FACTOR
C
   IF(NMATS.NE.1) GO TO 25
19  DO 20 I=1,2
20  COEFE(I)=5./6.
   GO TO 27
25  DO 26 IELEM=1,NELEM
   KOUNT=0
   DO 26 ILAYR=2,NLAYR
   IF(MATNO(IELEM,ILAYR).EQ.MATNO(IELEM,ILAYR-1)) GO TO 26
   KOUNT=KOUNT+1
26  CONTINUE
   IF(KOUNT.EQ.0) GO TO 19
   CALL SHEARC(MATNO,MELEM,MLAYR,PROPS,MMATS,NPROP,
              COEFE,NLAYR,DMATT)
27  DO 28 IMATS=1,NMATS
   DMATT(4,4,IMATS)=DMATT(4,4,IMATS)*COEFE(1)
28  DMATT(5,5,IMATS)=DMATT(5,5,IMATS)*COEFE(2)
   WRITE(6,900) (COEFE(I),I=1,2)
900  FORMAT(/' COEFE(1)=' ,E15.8,5X,'COEFE(2)=' ,E15.8/)
C
   DO 80 IMATS=1,NMATS
C
C IF THE REFERENTIAL SYSTEM OF AXES COINCIDES WITH THE
C PRINCIPAL AXES OF MATERIAL -GOTO 80
C
   THETA=PROPS(IMATS,17)
   IF(ABS(THETA).LT.0.001) GO TO 80
C
C SET MATRIK TRANSFORMASI
C
   DO 30 I=1,5
   DO 30 J=1,5
30  TRANS(I,J)=0.0
   C=COS(THETA)
   S=SIN(THETA)
   TRANS(1,1)=C*C
   TRANS(1,2)=S*S
   TRANS(2,1)=TRANS(1,2)
   TRANS(2,2)=TRANS(1,1)
   TRANS(1,3)=C*S

```



```

TRANS(3,1)=-2.0*TRANS(1,3)
TRANS(2,3)=-TRANS(1,3)
TRANS(3,2)=-TRANS(3,1)
TRANS(3,3)=TRANS(1,1)-TRANS(1,2)
TRANS(4,4)=C
TRANS(4,5)=S
TRANS(5,4)=-S
TRANS(5,5)=C
C
C HITUNG PERKALIAN MATRIK D DAN MATRIK T
C
DO 35 I=1,5
DO 35 J=1,5
GASHM(I,J)=0.0
DO 35 K=1,5
35 GASHM(I,J)=GASHM(I,J)+DMATT(I,K,IMATS)*TRANS(K,J)
C
C HITUNG TRANSFORMASI DARI MATRIK D
C
DO 40 I=1,5
DO 40 J=1,5
DMATT(I,J,IMATS)=0.0
DO 40 K=1,5
40 DMATT(I,J,IMATS)=DMATT(I,J,IMATS)+TRANS(K,I)*GASHM(K,J)
DO 45 I=1,5
DO 45 J=1,5
45 DMATT(J,I,IMATS)=DMATT(I,J,IMATS)
C
C SET MATRIK PARAMETER ANISOTROPIK UNTUK MATERIAL
C
DO 50 I=1,5
DO 50 J=1,5
50 APARA(I,J)=0.0
APARA(1,1)=AMATX(1,IMATS)
APARA(1,2)=AMATX(2,IMATS)
APARA(2,1)=APARA(1,2)
APARA(2,2)=AMATX(4,IMATS)
APARA(3,3)=AMATX(6,IMATS)
APARA(4,4)=AMATX(7,IMATS)
APARA(5,5)=AMATX(9,IMATS)
C
C SET MATRIK TRANSFORMASI BARU
C
TRANS(3,1)=-C*S
TRANS(2,3)=2.0*TRANS(3,1)
TRANS(3,2)=-TRANS(3,1)
TRANS(1,3)=-TRANS(2,3)
C
C HITUNG PERKALIAN MATRIK A DAN MATRIK T
C
DO 55 I=1,5
DO 55 J=1,5
GASHM(I,J)=0.0
DO 55 K=1,5
55 GASHM(I,J)=GASHM(I,J)+APARA(I,K)*TRANS(K,J)
C
C HITUNG PARAMETER ANISOTROPIK YANG BARU
C
DO 60 I=1,5
DO 60 J=1,5
APARA(I,J)=0.0
DO 60 K=1,5
60 APARA(I,J)=APARA(I,J)+TRANS(K,I)*GASHM(K,J)
AMATX(1,IMATS)=APARA(1,1)
AMATX(2,IMATS)=APARA(1,2)

```

```

      AMATX(3,IMATS)=APARA(1,3)
      AMATX(4,IMATS)=APARA(2,2)
      AMATX(5,IMATS)=APARA(2,3)
      AMATX(6,IMATS)=APARA(3,3)
      AMATX(7,IMATS)=APARA(4,4)
      AMATX(8,IMATS)=APARA(4,5)
      AMATX(9,IMATS)=APARA(5,5)
80  CONTINUE
      RETURN
      END

      SUBROUTINE SHEARC(MATNO,MELEM,MLAYR,PROPS,MMATS,NPROP,
        COEFE,NLAYR,DMATT)
C
C  SUBROUTINE UNTUK MENGHITUNG SHEAR CORRECTION FAKTOR UNTUK
C  KASUS STRUKTUR KOMPOSIT LAMINASI
C
      DIMENSION RFACT(2),TRLOW(2),UPTER(2),GBARF(2),MATNO(MELEM,MLAYR),
        COEFE(2),ZETA1(2),ZETA2(2),DINDX(2),PROPS(MMATS,NPROP),
        GINDX(2),DIFF2(2),DIFF3(2),SUMLA(2),DMATT(5,5,MMATS),
        DIFF5(2)
C
C  INITIALISE SOME ARRAYS
C
      DO 10 I=1,2
        SUMLA(I)=0.0
        RFACT(I)=0.0
        GBARF(I)=0.0
        UPTER(I)=0.0
        TRLOW(I)=0.0
      10  COEFE(I)=0.0
C
C  HITUNG POSISI NETRAL AXIS
C
      DSUMM = 0.0
      DO 15 ILAYR = 1,NLAYR
        LPROP = MATNO(1,ILAYR)
        DZETA = PROPS(LPROP,3)
        ZHEIG = DSUMM+DZETA/2.
        DO 14 I = 1,2
          DINDX(I) = DMATT(I,I,LPROP)
          UPTER(I) = UPTER(I)+DINDX(I)*ZHEIG*DZETA
        14  TRLOW(I) = TRLOW(I)+DINDX(I)*DZETA
        15  DSUMM = DSUMM+DZETA
        DO 16 I = 1,2
          16  ZETA2(I) = -UPTER(I)/TRLOW(I)
C
C  HITUNG SHEAR CORRECTION FACTOR
C
      DO 20 ILAYR = 1,NLAYR
        LPROP = MATNO(1,ILAYR)
        DIFF1 = PROPS(LPROP,3)
        INDEX = 10
        DO 20 I = 1,2
          ZETA1(I) = ZETA2(I)
          ZETA2(I) = ZETA1(I)+DIFF1
          DIFF2(I) = ZETA2(I)**2-ZETA1(I)**2
          DIFF3(I) = ZETA2(I)**3-ZETA1(I)**3
          DIFF5(I) = ZETA2(I)**5-ZETA1(I)**5
C
          DINDX(I) = DMATT(I,I,LPROP)
          GINDX(I) = PROPS(LPROP,INDEX)
C
          RFACT(I) = RFACT(I)+DINDX(I)*DIFF3(I)/3.
          GBARF(I) = GBARF(I)+GINDX(I)*DIFF1/2.

```



```

      TERM1 = SUMLA(I)*SUMLA(I)*DIFF1
      TERM2 = DINDX(I)*(ZETA1(I)**4)*DIFF1/4.
      TERM3 = DINDX(I)*DIFF5(I)/20.
      TERM4 = -DINDX(I)*ZETA1(I)*ZETA1(I)*DIFF3(I)/6.
      TERM5 = SUMLA(I)*ZETA1(I)*ZETA1(I)*DIFF1
      TERM6 = -SUMLA(I)*DIFF3(I)/3.
      COEFE(I) = COEFE(I)+(TERM1+DINDX(I)*(TERM2+
        TERM3+TERM4+TERM5+TERM6))/GINDX(I)
      INDEX = INDEX+1
      SUMLA(I) = SUMLA(I)-DINDX(I)*DIFF2(I)/2.
20  CONTINUE
C
      DO 30 I= 1,2
30  COEFE(I) = RFACT(I)*RFACT(I)/(2.*GBARF(I)*COEFE(I))
C
      RETURN
      END

      SUBROUTINE WORKS(COORD,DICOS,LNODS,THICK,MELEM,MPOIN,
        NPOIN,M3POI)
C
C  SUBROUTINE UNTUK MENETAPKAN SISTEM AXIS ORTHOGONAL DAN
C  KETEBALAN DARI TIAP NODAL POINT
C
      DIMENSION COORD(MPOIN,8),LNODS(MELEM,9),THICK(MPOIN),
        DICOS(3,M3POI)
      COMMON WORMX(3,24),QVALU,DJACB
C
C  SET KOORDINAT ATAS DAN BAWAH
C
      DO 30 IPOIN=1,NPOIN
      DO 10 I=1,3
      WORMX(I,1)=COORD(IPOIN,I)
10  WORMX(I,2)=COORD(IPOIN,I+4)
      NGASH=3
      NGISH=NGASH+2
      QVALU=-1.0
C  SET VEKTOR V-3
      CALL VECT(1,2,NGISH,5)
C  SET HASIL PERKALIAN SKALAR DARI VEKTOR (V-3)*(V-3)
      CALL SINGOP(NGISH,4)
      THICK(IPOIN)=SQRT(QVALU)
C  CREATES AND NORMALIZES AT EACH NODE THE VECTOR V-1,V-2 AND V-3
      CALL FRAME(NGASH,NGISH,0,1)
C
C  SET UP THE DIRECTION COSSINE MATRIX OF THE LOCAL AXES AT EACH POINT
C  IN ORDER V-1, V-2, V-3
C
      NPOSI=(IPOIN-1)*3
      DO 20 I=1,3
      DO 20 J=1,3
      JPOSI=NPOSI+I
20  DICOS(J,JPOSI)=WORMX(J,I+2)
30  CONTINUE
      RETURN
      END

      SUBROUTINE VECT(N1,N2,N3,NOPN)
C
C  VECTOR MANIPULATIONS
C
      COMMON WORMX(3,24),QVALU,DJACB
      I1 = N1
      GO TO (1,2,3,4,5),NOPN

```

```

1 I1 = N2
2 QVALU = 0.0
  DO 10 I = 1,3
10 QVALU = QVALU + WORMX(I,N1)*WORMX(I,I1)
  GO TO (15,16),NOPN
16 IF (QVALU.NE.0.0) GO TO 18
  WRITE (6,17)
17 FORMAT (12H NULL VECTOR)
  STOP
C EXECUTION IS TERMINATED WHEN A VECTOR IS NULL
18 QVALU = 1.0/SQRT(QVALU)
  3 DO 12 I = 1,3
12 WORMX(I,N2) = WORMX(I,N1)*QVALU
  RETURN
  4 K = 3
    DO 13 I = 1,3
      J = 6 - I - K
      WORMX(I,N3) = WORMX(J,N1)*WORMX(K,N2) - WORMX(K,N1)*WORMX(J,N2)
13 K = 1
  RETURN
  5 DO 14 I = 1,3
14 WORMX(I,N3) = WORMX(I,N1) + QVALU*WORMX(I,N2)
15 RETURN
  END

```

SUBROUTINE SINGOP(N1,NOPN)

C

C VECTOR OR MATRIX MANIPULATIONS

C

```

COMMON WORMX(3,24),QVALU,DJACB
GO TO (1,2,3,4,5),NOPN
1 CALL VECT(N1,N1,0,3)
  RETURN
2 CALL VECT(N1,N1,0,2)
  RETURN
3 CALL MATM(N1,N1,0,5)
  RETURN
4 CALL VECT(N1,N1,0,1)
  RETURN
5 N2 = N1 + 2
  I1 = 0
  DO 12 J = N1,N2
    DO 11 I = 1,3
11 WORMX(I,J) = 0.0
    II = II + 1
12 WORMX(II,J) = 1.0
  RETURN
  END

```

SUBROUTINE FRAME(N1,N2,N3,NOPN)

C

C MULTIPLE VECTOR AND/OR MATRIX MANIPULATIONS

C

```

COMMON WORMX(3,24),QVALU,DJACB
N3 = N1 + 2
I2 = N2 - 1
IF (I2.GE.N1) GO TO 10
I2 = I2 + 3
10 I1 = N1 + N1 + N1 + 3 - N2 - I2
  GO TO (1,2,3,4,5),NOPN
  1 WORMX(1,I1)=WORMX(3,N2)
    WORMX(2,I1)=0.0
    WORMX(3,I1)=-WORMX(1,N2)
    IF (WORMX(1,I1).EQ.0.0.AND.WORMX(3,I1).EQ.0.0)
      .WORMX(1,I1)=-WORMX(2,N2)

```



```

      CALL VECT (N2,I1,I2,4)
      GO TO 14
2    CALL MATM (I1,I2,0,7)
      CALL VECT (I1,I2,N2,4)
14   CALL MATM (N1,N1,0,6)
      RETURN
3    I1 = N1 + 1
      I2 = M3
      DO 11 I = 1,50
      DO 11 N = N1, M3
      CALL MATM (I1,I2,0,7)
      I1 = I2
11   I2 = N
      RETURN
4    CALL MATM (N1,N2,N3,2)
      CALL MATM (N3,N1,N2,3)
      RETURN
5    CALL MATM (N1,N2,N3,3)
      CALL SINGOP (N3,3)
      CALL MATM (N1,N3,N2,3)
      RETURN
      END

      SUBROUTINE MATM(N1,N2,N3,NOPN)
C
C MATRIX MANIPULATIONS
C
      COMMON WORMX(3,24),QVALU,DJACB
      GO TO (1,2,3,4,5,6,7,8,9),NOPN
1    K = 2
      DO 10 I = 1,3
      J = 4 - I - K
      M1 = N1 + J
      M2 = N1 + K
      M3 = N2 + I - 1
      M4 = N1 + I - 1
      CALL VECT(M1,M2,M3,4)
      CALL VECT(M4,M3,0,1)
      IF (QVALU.NE.0.0) GO TO 22
      WRITE (6,21)
21   FORMAT (17H ZERO DETERMINAT)
      STOP
C EXECUTION IS TERMINATED WHEN THE DETERMINAT IS ZERO
22   QVALU = 1.0/QVALU
      CALL VECT(M3,M3,0,3)
10   K = I - 1
      RETURN
2    DO 11 I = 1,3
      M1 = N1 + I - 1
      DO 11 J = 1,3
      M2 = N2 + J - 1
      M3 = N3 + J - 1
      CALL VECT(M1,M2,0,1)
11   WORMX(I,M3) = QVALU
      RETURN
3    DO 13 I = 1,3
      DO 13 K = 1,3
      M2 = N2 + K - 1
      M3 = N3 + K - 1
      GASH = 0.0
      DO 12 L = 1,3
      M1 = N1 + L - 1
12   GASH = GASH + WORMX(I,M1)*WORMX(L,M2)
13   WORMX(I,M3) = GASH
      RETURN

```

```

4 DO 14 I = 1,3
  M1 = N1 + I - 1
  CALL VECT(M1,N2,0,1)
14 WORMX(I,N3) = QVALU
  RETURN
5 DO 15 I = 1,3
  N1I = N1 + I - 1
  N2I = N2 + I - 1
  DO 15 J = 1,3
    N1J = N1 + J - 1
    N2J = N2 + J - 1
    GASH = WORMX(J,N1I)
    WORMX(J,N2I) = WORMX(I,N1J)
15 WORMX(I,N2J) = GASH
  RETURN
6 DO 16 I = 1,3
  I1 = N1 + I - 1
  I2 = N2 + I - 1
16 CALL VECT(I1,I2,0,2)
  RETURN
7 CALL SINGOP(N1,2)
  CALL SINGOP(N2,2)
  CALL VECT(N1,N2,0,1)
  GASH = -QVALU/(1.0 + SQRT(1.0-QVALU*QVALU))
  DO 17 I = 1,3
    GISH = WORMX(I,N1)
    GOSH = WORMX(I,N2)
    WORMX(I,N1) = GISH + GASH*GOSH
17 WORMX(I,N2) = GOSH + GASH*GISH
  RETURN
8 DO 18 J = 1,3
  N1J = N1 + J - 1
  N2J = N2 + J - 1
  DO 18 I = 1,3
18 WORMX(I,N2J) = WORMX(I,N1J)
  RETURN
9 DO 20 I = 1,3
  GASH = 0.0
  DO 19 J = 1,3
    N1J = N1 + J - 1
19 GASH = GASH + WORMX(I,N1J)*WORMX(J,N2)
20 WORMX(I,N3) = GASH
  RETURN
  END

  SUBROUTINE BGMAT(COORD,DICOS,LNODS,MATNO,MELEM,
    . MLAYR,MMATS,MPOIN,M3POI,NELEM,
    . NEVAB,NGAUS,NGAUZ,NLAYR,NNODE,NPROP,
    . POSGP,PROPS,THICK,WEIGP)
C
C SUBROUTINE UNTUK MENGHITUNG MATRIK B DAN G
C
  COMMON WORMX(3,24),QVALU,DJACB
  DIMENSION BMATX(5,45),BDUMY(8,45),COORD(MPOIN,8),
    . DICOS(3,M3POI),FUNCT(4),GMATX(2,45),
    . LNODS(MELEM,9),MATNO(MELEM,MLAYR),POSGP(5),
    . PROPS(MMATS,NPROP),THICK(MPOIN),SHAPE(3,9),
    . WEIGP(5)
  REWIND 8
  LGAUS=NGAUS-NGAUZ
C
C LGAUS=0 FOR NORMAL OR REDUCED INTEGRATION RULE, LGAUS=1 FOR
C SELECTIVE INTEGRATION RULE
C
  DO 100 IELEM=1,NELEM

```



```

      IF(LGAUS.EQ.0) GO TO 25
      NBORP=0
C
C REDUCED INTEGRATION
C
      CONSl=1.0/POSGP(4)
      CONS2=-CONSl
      ZETSP=0.0
      KGAUZ=-1
      DO 20 IGAUZ=1,NGAUZ
      DO 20 JGAUZ=1,NGAUZ
      KGAUZ=KGAUZ+1
      EXISP=POSGP(3+IGAUZ)
      ETASP=POSGP(3+JGAUZ)
      CALL      SFRI(SHAPE,EXISP,ETASP)
      CALL      FUNC(BMATX,SHAPE,THICK,NBORP,NNODE,ZETSP,MELEM,
        .        COORD,DICOS,LNODS,IELEM,MPOIN,M3POI,GMATX)
      DO 15 IEVAB=1,NEVAB
      DO 15 IPOSI=1,2
      JPOSI=2*KGAUZ+IPOSI
15  BDUMY(JPOSI,IEVAB)=GMATX(IPOSI,IEVAB)
20  CONTINUE
C
C SET MATRIK B DAN MATRIK G UNTUK NORMAL DAN REDUCED INTEGRATION RULE
C
25  NBORP=1
      DO 50 IGAUS=1,NGAUS
      DO 50 JGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)
      CALL      SFRI(SHAPE,EXISP,ETASP)
      ZETSP=-1.0
      DO 45 ILAYR=1,NLAYR
      LPROP=MATNO(IELEM,ILAYR)
      DZETA=PROPS(LPROP,3)
      ZETSP=ZETSP+DZETA/2.0
      CALL      FUNC(BMATX,SHAPE,THICK,NBORP,NNODE,ZETSP,MELEM,
        .        COORD,DICOS,LNODS,IELEM,MPOIN,M3POI,GMATX)
      DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)*DZETA
      IF(LGAUS.EQ.0) GO TO 40
C
C SELECTIVE INTEGRATION
C
      FUNCT(1)=0.25*(1.0+CONSl*EXISP)*(1.0+CONSl*ETASP)
      FUNCT(2)=0.25*(1.0+CONSl*EXISP)*(1.0+CONS2*ETASP)
      FUNCT(3)=0.25*(1.0+CONS2*EXISP)*(1.0+CONSl*ETASP)
      FUNCT(4)=0.25*(1.0+CONS2*EXISP)*(1.0+CONS2*ETASP)
C
      DO 30 IEVAB=1,NEVAB
      DO 30 IDOFN=4,5
      BMATX(IDOFN,IEVAB)=0.0
      DO 30 INTPO=1,4
      IGASH=2*INTPO+IDOFN-5
30  BMATX(IDOFN,IEVAB)=BMATX(IDOFN,IEVAB)+FUNCT(INTPO)*
        .        BDUMY(IGASH,IEVAB)
40  CONTINUE
      WRITE(8) BMATX,GMATX,DVOLU
      ZETSP=ZETSP+DZETA/2.0
45  CONTINUE
50  CONTINUE
100 CONTINUE
      RETURN
      END

      SUBROUTINE FUNC(BMATX,SHAPE,THICK,NBORP,NNODE,ZETA,MELEM,

```

```

      COORD, DICOS, LNODS, IELEM, MPOIN, M3POI, GMATX)
C
C SUBROUTINE UNTUK MENENTUKAN MATRIK B DAN NILAI JACOBIAN
C
      COMMON WORMX(3,24), QVALU, DJACB
      DIMENSION BMATX(5,45), SHAPE(3,9), THICK(MPOIN), GMATX(2,45),
      COORD(MPOIN,8), DICOS(3,M3POI), LNODS(MELEM,9)
C
C MENENTUKAN J-TRANSPOSE
C
      DO 20 I = 1,3
      DO 20 J = 1,4
      20 WORMX(I,J)=0.0
C UNTUK ELEMEN DENGAN 8 NODE POINT
      DO 24 INODE = 1,8
      IPOIN=IABS(LNODS(IELEM, INODE))
      DO 24 K = 1,3
      GTOP = COORD(IPOIN,K)
      GBOT = COORD(IPOIN,K+4)
      GOSH = ((1.0+ZETA)*GTOP + (1.0-ZETA)*GBOT)/2.0
      DO 22 J = 1,3
      22 WORMX(K,J) = WORMX(K,J) + GOSH*SHAPE(J, INODE)
      24 WORMX(K,4) = WORMX(K,4) + SHAPE(1, INODE)*(GTOP-GBOT)/2.0
C MENENTUKAN J-INVERSE
      CALL MATM (2,5,0,1)
      DJACB = 1.0/QVALU
C
      IF (NBORP.EQ.2) GO TO 50
C CREATES DIRECTION ZETA NORMAL TO XI AND ETA
      CALL VECT (2,3,10,4)
C CREATES LOCAL CARTESIAN SET
      CALL FRAME (8,10,0,1)
      IF (NBORP.EQ.4) GO TO 50
C
C MENENTUKAN MATRIK STRAIN/DISPLACEMENT
C
      DO 40 INODE = 1, NNODE
      DO 40 J = 1,5
C
      NSHAP = 5*(INODE-1) + J
      DO 26 I = 11,17
      DO 26 K = 1,3
      26 WORMX(K,I) = 0.0
      IF (J.GE.4) GO TO 30
      DO 29 K = 1,3
C
      29 WORMX(J,K+10) = SHAPE(K, INODE)
      GO TO 36
      30 JPOSI = J-3
C
      DO 35 M = 1,3
C
      IPOIN=IABS(LNODS(IELEM, INODE))
      IPOSI=(IPOIN-1)*3
      GASH=DICOS(M, IPOSI+JPOSI)
      IF (JPOSI.NE.2) GO TO 32
      GASH = -GASH
      32 DO 34 K = 1,4
C
      IF (K.EQ.4) GO TO 31
      WORMX(M, K+10) = ZETA*SHAPE(K, INODE)*GASH*(THICK(IPOIN)/2.)
C
      GO TO 34
      31 WORMX(M, K+10) = SHAPE(1, INODE)*GASH*(THICK(IPOIN)/2.)
      34 CONTINUE

```



```

35 CONTINUE
C
36 CALL SINGOP (12,3)
C
CALL MATM (5,12,15,3)
C
CALL MATM (15,18,0,8)
C
CALL FRAME (8,18,21,4)
C
IF(NBORG.EQ.0) GO TO 39
BMATX(1,NSHAP) = WORMX(1,18)
BMATX(2,NSHAP) = WORMX(2,19)
BMATX(3,NSHAP) = WORMX(2,18) + WORMX(1,19)
BMATX(4,NSHAP) = WORMX(1,20) + WORMX(3,18)
BMATX(5,NSHAP) = WORMX(2,20) + WORMX(3,19)
C
GMATX(1,NSHAP)=WORMX(1,20)
GMATX(2,NSHAP)=WORMX(2,20)
C
GO TO 40
39 CONTINUE
C
GMATX(1,NSHAP)=WORMX(1,20)+WORMX(3,18)
GMATX(2,NSHAP)=WORMX(2,20)+WORMX(3,19)
40 CONTINUE
50 RETURN
END

```

```

SUBROUTINE SFR1(W,G,H)
C
C SUBROUTINE UNTUK MENGEVALUASI SHAPE FUNCTION
C

```

```

DIMENSION W(3,9)
GG = G*G
GH = G*H
HH = H*H
GGH = GG*H
GHH = G*HH
G2 = G*2
H2 = H*2
GH2 = GH*2
W(1,1) = (-1.+GH+GG+HH-GGH-GHH)/4.
W(1,2) = ( 1.-H-GG+GGH)/2.
W(1,3) = (-1.-GH+GG+HH-GGH+GHH)/4.
W(1,4) = ( 1.+G-HH-GHH)/2.
W(1,5) = (-1.+GH+GG+HH+GGH+GHH)/4.
W(1,6) = ( 1.+H-GG-GGH)/2.
W(1,7) = (-1.-GH+GG+HH+GGH-GHH)/4.
W(1,8) = ( 1.-G-HH+GHH)/2.
W(1,9) = 1.0-GG-HH+GG*HH
W(2,1) = ( H+G2-GH2-HH)/4.
W(2,2) = -G+GH
W(2,3) = (-H+G2-GH2+HH)/4.
W(2,4) = ( 1.-HH)/2.
W(2,5) = ( H+G2+GH2+HH)/4.
W(2,6) = -G-GH
W(2,7) = (-H+G2+GH2-HH)/4.
W(2,8) = (-1.+HH)/2.
W(2,9) = -G2*(1.0-HH)
W(3,1) = ( G+H2-GG-GH2)/4.
W(3,2) = (-1.+GG)/2.
W(3,3) = (-G+H2-GG+GH2)/4.
W(3,4) = -H-GH
W(3,5) = ( G+H2+GG+GH2)/4.

```

```

W(3,6) = ( 1.-GG)/2.
W(3,7) = (-G+H2+GG-GH2)/4.
W(3,8) = -H+GH
W(3,9) = -H2*(1.0-GG)
RETURN
END

```

```

SUBROUTINE LOADS(ANVEL,COORD,ELOAD,GRAVI,LNODS,
.             MATNO,MELEM,MEVAB,MMATS,MPOIN,DICOS,
.             NELEM,NEVAB,NGAUS,THICK,
.             NNODE,NPROP,NSTRE,POSGP,M3POI,
.             PROPS,WEIGP,MLAYR,NLAYR)

```

```

C
C SUBROUTINE UNTUK MENENTUKAN NODAL FORCE
C

```

```

COMMON WORMX(3,24),QVALU,DJACE
DIMENSION BMATX(5,45),COORD(MPOIN,8),
.         ELOAD(MELEM,MEVAB),GRAVI(3),LNODS(MELEM,9),
.         MATNO(MELEM,MLAYR),POSGP(5),DICOS(3,M3POI),
.         PROPS(MMATS,NPROP),SHAPE(3,9),STREN(5),
.         THICK(MPOIN),WEIGP(5),
.         GMATX(2,45),NPRES(MELEM),NUCLO(MELEM),NBODY(MELEM)

```

```

REWIND 8

```

```

C
C LOOP OVER EACH ELEMENT
C

```

```

DO 999 ILOAD=1,NELEM
  READ(109,900)NPRES,NUCLO,NBODY
999 CONTINUE
DO 150 IELEM=1,NELEM

```

```

C
C BACA KARAKTERISTIK BEBAN YANG BEKERJA
C

```

```

  READ(109,900)NPRES(IELEM),NUCLO(IELEM),NBODY(IELEM)
900 FORMAT(5I5)
  WRITE(6,901) IELEM,NPRES(IELEM),NUCLO(IELEM),NBODY(IELEM)
901 FORMAT(8H NELEM =,I5,5X,8H NPRES =,I5,5X,8H NUCLO =,I5,
. 5X,8H NBODY =,I5)
  IF(NPRES(IELEM).EQ.0) GO TO 3
  READ(5,902) KPRES,CFACE,PRESA,SURFA
  WRITE(6,902) KPRES,CFACE,PRESA,SURFA
902 FORMAT(I5,F5.1,2F15.5)

```

```

C
  3 CONTINUE
C

```

```

C INITIALIZE THE LOAD MATRIX ELOAD ONE COLUMN AT EACH TIME
C

```

```

DO 4 IEVAB=1,NEVAB
  4 ELOAD(IELEM,IEVAB)=0.0

```

```

C
C ENTER LOOPS OVER GAUSS POINTS FOR NUMERICAL INTEGRATION
C

```

```

DO 145 IGAUS=1,NGAUS
DO 145 JGAUS=1,NGAUS
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
CALL SFRI(SHAPE,EXISP,ETASP)
IF(NBODY(IELEM).EQ.0) GO TO 141
ZETSP=-1.0
DO 140 ILAYR=1,NLAYR
LPROP=MATNO(IELEM,ILAYR)
DZETA=PROPS(LPROP,3)
ZETSP=ZETSP+DZETA/2.0
READ(8) BMATX,GMATX,DVOLUME

```



```

C
C HITUNG CENTRIFUGAL FORCE
C
      IF (ANVEL.EQ.0.0) GO TO 70
      NBORP = 2
      CALL      FUNC (BMATX, SHAPE, THICK, NBORP, NNODE, ZETSP,
                     MELEM, COORD, DICOS, LNODS, IELEM, MPOIN,
                     M3POI, GMATX)
      GASH=PROPS (LPROP, 4) *ANVEL*ANVEL*DVOLU
      DO 45 I=1, 2
45  STREN(I)=GASH*WORMX(I, 1)
      STREN(3)=0.0
      DO 65 INODE=1, NNODE
C
      IPOIN=IABS (LNODS (IELEM, INODE))
      JPOSI=(IPOIN-1)*3
      DO 65 ISTRE=1, NSTRE
      IEVAB=(INODE-1)*5+ISTRE
      IF (ISTRE.GT.3) GO TO 50
      ELOAD (IELEM, IEVAB)=ELOAD (IELEM, IEVAB)+STREN(ISTRE)*
                               SHAPE(1, INODE)
      GO TO 65
50  JPOSI=JPOSI+1
      GASH=SHAPE(1, INODE)*(THICK(IPOIN)/2.0)*ZETSP
      IF (ISTRE.NE.5) GO TO 55
      GASH=-GASH
55  DO 60 I=1, 2
60  ELOAD (IELEM, IEVAB)=ELOAD (IELEM, IEVAB)+STREN(I)*
                               DICOS(I, JPOSI)*GASH
      65 CONTINUE
      70 CONTINUE
C
C GRAVITY LOADS
C
      GASH=PROPS (LPROP, 4) *DVOLU
      DO 75 I=1, 3
75  STREN(I)=GRAVI(I)*GASH
      DO 95 INODE=1, NNODE
      IPOIN=IABS (LNODS (IELEM, INODE))
      JPOSI=(IPOIN-1)*3
      DO 95 ISTRE=1, NSTRE
      IEVAB=(INODE-1)*5+ISTRE
      IF (ISTRE.GT.3) GO TO 80
      ELOAD (IELEM, IEVAB)=ELOAD (IELEM, IEVAB)+STREN(ISTRE)*
                               SHAPE(1, INODE)
      GO TO 95
80  JPOSI=JPOSI+1
      GASH=SHAPE(1, INODE)*(THICK(IPOIN)/2.0)*ZETSP
      IF (ISTRE.NE.5) GO TO 85
      GASH=-GASH
85  DO 90 I=1, 3
90  ELOAD (IELEM, IEVAB)=ELOAD (IELEM, IEVAB)+STREN(I)*
                               DICOS(I, JPOSI)*GASH
      95 CONTINUE
      ZETSP=ZETSP+DZETA/2.0
140 CONTINUE
C
C NODAL LOADS DUE TO PRESSURE
C
141 IF (NPRES(IELEM).EQ.0) GO TO 142
      CALL      PRES (BMATX, COORD, ELOAD, LNODS, POSGP, SHAPE, THICK,
                     WEIGP, IELEM, IGAUS, JGAUS, MELEM, MPOIN, NNODE,
                     NEVAB, KPRES, CFACE, PREVA, SURFA, DICOS, M3POI)
142 CONTINUE
145 CONTINUE

```

```

C
C POINT LOADS
C
      IF (NUCLO(IELEM).EQ.0) GO TO 150
C
      DO 120 IGASP=1, NUCLO(IELEM)
      READ(5,950) LPOIN, LDOFN, CARGA
      WRITE(6,960) LPOIN, LDOFN, CARGA
      IEVAB=(LPOIN-1)*5+LDOFN
120  ELOAD(IELEM, IEVAB)=ELOAD(IELEM, IEVAB)+CARGA
150  CONTINUE
950  FORMAT(2I5, F10.5)
960  FORMAT(/8H LPOIN =, I5, 8H LDOFN =, I5, 7H LOAD =, F16.8/)
      RETURN
      END

      SUBROUTINE PRES(BMATX, COORD, ELOAD, LNODS, POSGP, SHAPE, THICK,
      WEIGP, IELEM, IGAUS, JGAUS, MELEM, MPOIN, NNODE,
      NEVAB, KPRES, CFACE, PREVA, SURFA, DICOS, M3POI)
C
C SUBROUTINE UNTUK MENGEVALUASI NODAL LOADS AKIBAT TEKanan
C
      COMMON WORMX(3,24), QVALU, DJACB
      DIMENSION BMATX(5,45), COORD(MPOIN,8), ELOAD(MELEM, NEVAB),
      LNODS(MELEM,9), POSGP(5), PREMXX(2,9), SHAPE(3,9),
      THICK(MPOIN), WEIGP(5), DICOS(3,M3POI), GMATX(2,45)
      ZETA=CFACE
      NBORP=2
      CALL      FUNC(BMATX, SHAPE, THICK, NBORP, NNODE, ZETA, MELEM,
      COORD, DICOS, LNODS, IELEM, MPOIN, M3POI, GMATX)
C
C HYDROSTATIC OR SPECIFIED AS NODAL COORDINATES
C
      IF(KPRES.EQ.0) GO TO 20
      IF(KPRES.EQ.2) GO TO 10
      WORMX(3,1)=WORMX(3,1)-SURFA
      PRESS=PREVA*WORMX(3,1)
      IF(PRESS.GE.0.0) GO TO 25
      PRESS=0.0
      GO TO 25
10  PREVA=0.0
      DO 15 INODE=1,8
      NGASH=LABS(LNODS(IELEM, INODE))
C
C SET UP ARRAY
C
      PREMXX(1, INODE)=COORD(NGASH,4)
      PREMXX(2, INODE)=COORD(NGASH,8)
      GISH=((1.0+ZETA)*PREMXX(1, INODE)+(1.0-ZETA)*PREMXX(2, INODE))/2.0
15  PREVA=PREVA+GISH*SHAPE(1, INODE)
20  PRESS=PREVA
25  GMULT=WEIGP(IGAUS)*WEIGP(JGAUS)*CFACE*PRESS
C
C HITUNG CONSISTENT NODAL LOADS
C
      DO 45 INODE=1, NNODE
      IPOIN=LABS(LNODS(IELEM, INODE))
      QVALU=-GMULT*SHAPE(1, INODE)*DJACB
      CALL      VECT(7,21,0,3)
      DO 30 I=1,3
      IPOSI=(INODE-1)*5+I
30  ELOAD(IELEM, IPOSI)=ELOAD(IELEM, IPOSI)+WORMX(I,21)
      QVALU=ZETA*THICK(IPOIN)/2.0
      CALL      SINGOP(21,1)
      NPOSI=(IPOIN-1)*3

```



```

DO 40 I=1,2
  JPOSI=(INODE-1)*5+(I+3)
  NPOSI=NPOSI+1
  DO 32 K=1,3
32  WORMX(K,24)=DICOS(K,NPOSI)
    CALL      VECT(21,24,0,1)
    IF(I.EQ.2) GO TO 35
    QVALU=-QVALU
35  ELOAD(IELEM,JPOSI)=ELOAD(IELEM,JPOSI)+QVALU
40  CONTINUE
45  CONTINUE
    RETURN
    END

    SUBROUTINE ZERO(EFFST,ELOAD,EPSTN,MELEM,MEVAB,KINCS,
.          MTOTG,MTOTV,NDOFN,NELEM,NEVAB,NREST,
.          NSTRE,NTOTG,NTOTV,MVFIX,MVFIX,STRSG,
.          TDISP,TFACT,TLOAD,TREAC)

C
C  SUBROUTINE UNTUK MENGINTIAL KESELURUH ARRAY MENJADI NOL
C
    DIMENSION ELOAD(MELEM,MEVAB),STRSG(5,MTOTG),TDISP(MTOTV),
.          TLOAD(MELEM,MEVAB),TREAC(MVFIX,5),EPSTN(MTOTG),
.          EFFST(MTOTG)
    IF(NREST.EQ.1) GO TO 70
    KINCS=0
    TFACT=0.0
    DO 30 IELEM=1,NELEM
    DO 30 IEVAB=1,NEVAB
    ELOAD(IELEM,IEVAB)=0.0
30  TLOAD(IELEM,IEVAB)=0.0
    DO 40 ITOTV=1,NTOTV
40  TDISP(ITOTV)=0.0
    DO 50 IVFIX=1,MVFIX
    DO 50 IDOFN=1,NDOFN
50  TREAC(IVFIX,IDOFN)=0.0
    DO 60 ITOTG=1,NTOTG
    EPSTN(ITOTG)=0.0
    EFFST(ITOTG)=0.0
    DO 60 ISTR1=1,NSTRE
60  STRSG(ISTR1,ITOTG)=0.0
    GO TO 80
70  REWIND 12
    READ(12) KINCS,TFACT,EFFST,ELOAD,EPSTN
    READ(12) TDISP,TLOAD,TREAC,STRSG
80  CONTINUE
    RETURN
    END

    SUBROUTINE INCREM(ELOAD,FIXED,IINCS,MELEM,MEVAB,MITER,
.          MTOTV,MVFIX,NDOFN,NELEM,NEVAB,NOUTP,
.          NOFIX,NTOTV,MVFIX,PRESG,RLOAD,TFACT,
.          TLOAD,TOLER,LNODS,IFFIX,NNODE,NCOLA,
.          NREST,KINCS)

C
C  SUBROUTINE UNTUK MENENTUKAN FAKTOR INCREMENT PEMBEBANAN
C
    DIMENSION ELOAD(MELEM,MEVAB),FIXED(MTOTV),
.          NOUTP(2),NOFIX(MVFIX),PRESG(MVFIX,NDOFN),
.          RLOAD(MELEM,MEVAB),TLOAD(MELEM,MEVAB),
.          LNODS(MELEM,9),IFFIX(MTOTV)
    OPEN(110,FILE='C:\DATA\OUTP.DAT')
    IF(NREST.EQ.0) GO TO 20
    IF(IINCS.GT.KINCS) GO TO 20
    DO 10 IINCS=1,KINCS

```

```

10 READ(110,950) FACTO
20 WRITE(6,900) IINCS
900 FORMAT(///,5X,22H*** INCREMENT NUMBER ,I5)
    READ(110,950) FACTO,TOLER,MITER,NOUTP(1),NOUTP(2)
950 FORMAT(2F10.5,3I5)
    TFACT=TFACT+FACTO
    WRITE(6,960) TFACT,TOLER,MITER,NOUTP(1),NOUTP(2)
960 FORMAT(1H0,5X,13HLOAD FACTOR =,F10.5,5X,
.24H CONVERGENCE TOLERANCE =,F10.5,5X,24HMAX. NO. OF ITERATIONS =,
. I5,//32H INITIAL OUTPUT PARAMETER =,I5,5X,24HFINAL OUTPUT PA
. RAMETER =,I5)
    DO 80 IELEM=1,NELEM
    DO 80 IEVAB=1,NEVAB
    ELOAD(IELEM,IEVAB)=ELOAD(IELEM,IEVAB)+RLOAD(IELEM,IEVAB)*FACTO
80 TLOAD(IELEM,IEVAB)=TLOAD(IELEM,IEVAB)+RLOAD(IELEM,IEVAB)*FACTO
C
C INTERPRET FIXITY DATA IN VECTOR FORM
C
    DO 100 ITOTV=1,NTOTV
100 FIXED(ITOTV)=0.0
    DO 110 IVFIX=1,NVFIX
    NLOCA=(NOFIX(IVFIX)-1)*NDOFN
    DO 110 IDOFN=1,NDOFN
    NGASH=NLOCA+IDOFN
    FIXED(NGASH)=PRESK(IVFIX,IDOFN)*FACTO
110 CONTINUE
C
C ADDITIONAL CONSTRAINTS FOR THE HETEROSIS ELEMENT
C
    IF(NNODE.EQ.8.OR.NCOLA.NE.1) GO TO 130
    DO 120 IELEM=1,NELEM
    LNOD9=LNODS(IELEM,9)
    NLOCA=(LNOD9-1)*NDOFN
C
    IDOFN=3
    DO 120 IDOFN=1,3
    NGASH=NLOCA+IDOFN
120 IFFIX(NGASH)=-1
130 CONTINUE
    RETURN
    END

    SUBROUTINE ALGOR(FIXED,KITER,IITER,KRESL,MTOTV,NALGO,
NTOTV,KUNLO,KINCS)
C
C SUBROUTINE UNTUK MENETAPKAN RESOLUSI PERSAMAAN
C
    DIMENSION FIXED(MTOTV)
    KRESL=2
    IF(NALGO.EQ.1.AND.KITER.EQ.2) KRESL=1
    IF(NALGO.EQ.2) KRESL=1
    IF(NALGO.EQ.3.AND.IITER.EQ.1) KRESL=1
    IF(NALGO.EQ.4.AND.KITER.EQ.2) KRESL=1
    IF(NALGO.EQ.4.AND.IITER.EQ.2) KRESL=1
    IF(NALGO.EQ.4.AND.KUNLO.GT.0) KRESL=1
    IF(KITER.EQ.(KINCS+1+1)) KRESL=1
    IF(IITER.EQ.1) RETURN
    DO 100 ITOTV = 1,NTOTV
    FIXED(ITOTV)=0.0
100 CONTINUE
    RETURN
    END

    SUBROUTINE STIFF(EPSTN,ESTIF,KITER,LNODS,MATNO,
MELEM,MEVAB,MMATS,MPOIN,MTOTG,NDOFN,
NELEM,NEVAB,NGAUS,NNODE,NPROP,

```



```

      NSTRE, POSGP, PROPS, STRSG, WEIGP, AMATX,
      DMATT, MLAYR, NLAYR, THICK,
      TDISP, MTOTV, LARGE)
C
C SUBROUTINE UNTUK MENGEVALUASI MATRIK KEKAKUAN DARI
C TIAP ELEMEN
C
      DIMENSION BMATX(5,45), DBMAT(5,45), DMATX(5,5),
      ESTIF(MEVAB,MEVAB), LNODS(MELEM,9), MATNO(MELEM,MLAYR),
      POSGP(5), PROPS(MMATS,NPROP), AMATX(9,MMATS),
      STRES(5), THICK(MPOIN), WEIGP(5), EPSTN(MTOTG),
      STRSG(5,MTOTG), AVECT(5), DVECT(5), DMATT(5,5,MMATS),
      TDISP(MTOTV), ETDIS(45), GMATX(2,45)
      REWIND 1
      REWIND 8
      KGAUS=0
C
C LOOP OVER EACH ELEMEN
C
      DO 110 IELEM=1, NELEM
C SET UP THE ELEMENT DISPLACEMENT VECTOR
      JPOSI=0
      DO 10 INODE=1, NNODE
      LNODE=IABS(LNODS(IELEM, INODE))
      NPOSN=(LNODE-1)*NDOFN
      DO 10 IDOFN=1, NDOFN
      NPOSN=NPOSN+1
      JPOSI=JPOSI+1
      10 ETDIS(JPOSI)=TDISP(NPOSN)
C
C INITIALIZE THE ELEMENT STIFFNESS MATRIX
C
      DO 20 IEVAB=1, NEVAB
      DO 20 JEVAB=1, NEVAB
      20 ESTIF(IEVAB, JEVAB)=0.0
      KGASP=0
C
C ENTER LOOPS OVER GAUSS POINTS FOR NUMERICAL INTEGRATION
C
      DO 105 IGAUS=1, NGAUS
      DO 105 JGAUS=1, NGAUS
      DO 100 ILAYR=1, NLAYR
      LPROP=MATNO(IELEM, ILAYR)
      KGASP=KGASP+1
      KGAUS=KGAUS+1
      READ(8) BMATX, GMATX, DVOLU
C SET MATRIK B PADA ANALISA LARGE DISPLASMENT
      IF(LARGE.EQ.1.AND.KITER.GT.2)
      .CALL      LDISP(BMATX, GMATX, ETDIS, NEVAB)
      IF(KITER.EQ.2) GO TO 80
      IF(EPSTN(KGAUS).LE.0.0) GO TO 80
C
C HITUNG MATRIK ELASTO-PLASTIC /MATRIK D
C
      DO 50 ISTRE=1, NSTRE
      50 STRES(ISTRE)=STRSG(ISTRE, KGAUS)
      CALL      FLOWS(ABETA, AVECT, DVECT, LPROP,
      .          MMATS, NPROP, PROPS, STRES, AMATX, DMATT)
      DO 70 ISTRE=1, NSTRE
      DO 70 JSTRE=1, NSTRE
      70 DMATX(ISTRE, JSTRE)=DMATT(ISTRE, JSTRE, LPROP)-ABETA*
      DVECT(ISTRE)*DVECT(JSTRE)
C
C HITUNG PERKALIAN MATRIK D DAN MATRIK B
C

```

```

DO 35 ISTRE=1, NSTRE
DO 35 IEVAB=1, NEVAB
DBMAT(ISTRE, IEVAB)=0.0
DO 35 JSTRE=1, NSTRE
DBMAT(ISTRE, IEVAB)=DBMAT(ISTRE, IEVAB)+DMATX(ISTRE, JSTRE) *
      BMATX(JSTRE, IEVAB)
35 CONTINUE
GO TO 90
80 CONTINUE
DO 85 ISTRE=1, NSTRE
DO 85 IEVAB=1, NEVAB
DBMAT(ISTRE, IEVAB)=0.0
DO 85 JSTRE=1, NSTRE
85 DBMAT(ISTRE, IEVAB)=DBMAT(ISTRE, IEVAB) +
      DMATT(ISTRE, JSTRE, LPROP) * BMATX(JSTRE, IEVAB)
C
C HITUNG KEKAKUAN ELEMEN
C
90 DO 40 IEVAB=1, NEVAB
DO 40 JEVAB=IEVAB, NEVAB
DO 40 ISTRE=1, NSTRE
40 ESTIF(IEVAB, JEVAB)=ESTIF(IEVAB, JEVAB)+BMATX(ISTRE, IEVAB) *
      DBMAT(ISTRE, JEVAB) * DVOLU
IF (LARGE.EQ.0.OR.KITER.EQ.2) GO TO 98
C
C HITUNG MATRIK GEOMETRI
C
CALL GEOME(ESTIF, GMATX, STRSG, MEVAB, NEVAB, MTOTG,
      KGAUS, DVOLU)
98 CONTINUE
100 CONTINUE
105 CONTINUE
C
DO 60 IEVAB=1, NEVAB
DO 60 JEVAB=1, NEVAB
ESTIF(JEVAB, IEVAB)=ESTIF(IEVAB, JEVAB)
60 CONTINUE
C
C SIMPAN MATRIK KEKAKUAN TIAP ELEMEN
C
WRITE(1) ESTIF
110 CONTINUE
RETURN
END

SUBROUTINE LDISP(BMATX, GMATX, ETDIS, NEVAB)
C
C SUBROUTINE UNTUK MENGEVALUASI MATRIK INITIAL DISPLASMEN
C
DIMENSION BMATX(5, 45), GMATX(2, 45), ETDIS(45), ADUMM(3, 2),
      BLARG(3, 45)
C HITUNG X DAN Y DERIVATIVE DARI W DISPLACEMENT
DWDXX=0.0
DWDYY=0.0
DO 10 IEVAB=1, NEVAB
DWDXX=DWDXX+GMATX(1, IEVAB)*ETDIS(IEVAB)
10 DWDYY=DWDYY+GMATX(2, IEVAB)*ETDIS(IEVAB)
C
ADUMM(1, 1)=DWDXX
ADUMM(1, 2)=0.0
ADUMM(2, 1)=0.0
ADUMM(2, 2)=DWDYY
ADUMM(3, 1)=DWDYY
ADUMM(3, 2)=DWDXX
C

```



```

C HITUNG MATRIK LARGE DISPLASMEN /BLARG
C
  DO 20 IEVAB=1, NEVAB
  DO 20 I=1, 3
    BLARG(I, IEVAB)=0.0
    DO 20 J=1, 2
      20 BLARG(I, IEVAB)=BLARG(I, IEVAB)+ADUMM(I, J)*GMATX(J, IEVAB)
C
  DO 30 IEVAB=1, NEVAB
  DO 30 I=1, 3
    30 BMATX(I, IEVAB)=BMATX(I, IEVAB)+BLARG(I, IEVAB)
  RETURN
  END

  SUBROUTINE GEOME(ESTIF, GMATX, STRSG, MEVAB, NEVAB, MTOTG,
    KGAUS, DVOLU)
C
C SUBROUTINE UNTUK MENGHITUNG MATRIK GEOMETRI
C
  DIMENSION ESTIF(MEVAB, MEVAB), GMATX(2, 45), STRSG(5, MTOTG),
    GEMTX(45, 45), GDUMM(2, 45), STDUM(2, 2)
  DO 5 IEVAB=1, NEVAB
  DO 5 JEVAB=1, NEVAB
    5 GEMTX(IEVAB, JEVAB)=0.0
C
  STDUM(1, 1)=STRSG(1, KGAUS)
  STDUM(1, 2)=STRSG(3, KGAUS)
  STDUM(2, 1)=STRSG(3, KGAUS)
  STDUM(2, 2)=STRSG(2, KGAUS)
C
  DO 10 I=1, 2
  DO 10 IEVAB=1, NEVAB
    GDUMM(I, IEVAB)=0.0
    DO 10 J=1, 2
      10 GDUMM(I, IEVAB)=GDUMM(I, IEVAB)+STDUM(I, J)*GMATX(J, IEVAB)
C
C HITUNG MATRIK GEOMETRI
C
  DO 20 IEVAB=1, NEVAB
  DO 20 JEVAB=IEVAB, NEVAB
  DO 20 I=1, 2
    20 GEMTX(IEVAB, JEVAB)=GEMTX(IEVAB, JEVAB)+GMATX(I, IEVAB)*
      GDUMM(I, JEVAB)*DVOLU
C
  DO 30 IEVAB=1, NEVAB
  DO 30 JEVAB=IEVAB, NEVAB
    30 ESTIF(IEVAB, JEVAB)=ESTIF(IEVAB, JEVAB)+GEMTX(IEVAB, JEVAB)
  RETURN
  END

  SUBROUTINE FLOWS(ABETA, AVECT, DVECT, LPROP,
    MMATS, NPROP, PROPS, SG, A, DMATT)
C
C SUBROUTINE UNTUK MENGHITUNG FLOW VECTOR
C
  DIMENSION AVECT(5), DMATT(5, 5, MMATS), DVECT(5),
    PROPS(MMATS, NPROP), SG(5), A(9, MMATS)
C
  HARDS=PROPS(LPROP, 7)
C
  L=LPROP
  AFUNC=(A(1, L)*SG(1)*SG(1)+2.0*A(2, L)*SG(1)*SG(2)+2.0*A(3, L)*
    SG(1)*SG(3)+A(4, L)*SG(2)*SG(2)+2.0*A(5, L)*SG(2)*SG(3)+
    A(6, L)*SG(3)*SG(3)+A(7, L)*SG(4)*SG(4)+2.0*A(8, L)*SG(4)*
    SG(5)+A(9, L)*SG(5)*SG(5))*0.5

```

```

AVECT(1)=(A(1,L)*SG(1)+A(2,L)*SG(2)+A(3,L)*SG(3))/AFUNC
AVECT(2)=(A(2,L)*SG(1)+A(4,L)*SG(2)+A(5,L)*SG(3))/AFUNC
AVECT(3)=(A(3,L)*SG(1)+A(5,L)*SG(2)+A(6,L)*SG(3))/AFUNC
AVECT(4)=(A(7,L)*SG(4)+A(8,L)*SG(5))/AFUNC
AVECT(5)=(A(8,L)*SG(4)+A(9,L)*SG(5))/AFUNC
WRITE(6,910) AVECT
910 FORMAT(8H AVECT =,5E15.6)
C
DO 10 I=1,5
DVECT(I)=0.0
DO 10 J=1,5
10 DVECT(I)=DVECT(I)+DMATT(I,J,LPROP)*AVECT(J)
WRITE(6,920) DVECT
920 FORMAT(8H DVECT =,5E15.6)
C
DENOM=HARDS
DO 20 ISTR=1,5
20 DENOM=DENOM+AVECT(ISTR)*DVECT(ISTR)
ABETA=1.0/DENOM
WRITE(6,930) ABETA
930 FORMAT(8H ABETA =,E15.6)
RETURN
END

SUBROUTINE FRONT(ASDIS,ELOAD,EQRHS,EQUAT,ESTIF,FIXED,
.          GLOAD,GSTIF,IFFIX,IINCS,IITER,KRESL,
.          LOCEL,LNODS,MBUFA,MELEM,MEVAB,MFRON,
.          MSTIF,MTOTV,MVFIX,NACVA,NAMEV,NDEST,
.          NDOFN,NELEM,NEVAB,NNODE,NOFIX,NPIVO,
.          NPOIN,NTOTV,TDISP,TLOAD,TREAC,VECRV)
C
C SUBROUTINE UNTUK MENYELESAIKAN PERSAMAAN SIMULTAN
C DENGAN METODE FRONTAL
C
DIMENSION ASDIS(MTOTV),ELOAD(MELEM,MEVAB),EQRHS(MBUFA),
.          EQUAT(MFRON,MBUFA),ESTIF(MEVAB,MEVAB),FIXED(MTOTV),
.          GLOAD(MFRON),GSTIF(MSTIF),IFFIX(MTOTV),
.          LNODS(MELEM,9),LOCEL(MEVAB),NACVA(MFRON),
.          NAMEV(MBUFA),NDEST(MEVAB),NOFIX(MVFIX),
.          NPIVO(MBUFA),TDISP(MTOTV),TLOAD(MELEM,MEVAB),
.          TREAC(MVFIX,NDOFN),VECRV(MFRON)
NFUNC(I,J)=(J*J-J)/2+I
C
C CHANGE THE SIGN OF THE LAST APPEARANCE OF EACH NODE
C
IF(IINCS.GT.1.OR.IITER.GT.1) GO TO 455
DO 140 IPOIN=1,NPOIN
KLAST=0
DO 130 IELEM=1,NELEM
DO 120 INODE=1,NNODE
IF(LNODS(IELEM,INODE).NE.IPOIN) GO TO 120
KLAST=IELEM
NLAST=INODE
120 CONTINUE
130 CONTINUE
IF(KLAST.NE.0) LNODS(KLAST,NLAST)=-IPOIN
140 CONTINUE
455 CONTINUE
C
C START BY INITIALIZING EVERYTHING THAT MATTERS TO ZERO
C
DO 145 IBUFA=1,MBUFA
145 EQRHS(IBUFA)=0.0
DO 150 ISTIF=1,MSTIF
150 GSTIF(ISTIF)=0.0

```



```

DO 160 IFRON=1,MFRON
GLOAD(IFRON)=0.0
VECRV(IFRON)=0.0
NACVA(IFRON)=0
DO 160 IBUFA=1,MBUFA
160 EQUAT(IFRON,IBUFA)=0.0
C
C PREPARE FOR DISC READING AND WRITING OPERATIONS
C
      NBUFA=0
      IF(KRESL.GT.1) NBUFA=MBUFA
      REWIND 1
      REWIND 2
      REWIND 4
      REWIND 7
C
C ENTER MAIN ELEMEN ASSEMBLY-REDUCTION LOOP
C
      NFRON=0
      KELVA=0
      DO 320 IELEM=1,NELEM
      IF(KRESL.GT.1) GO TO 400
      KEVAB=0
      READ(1) ESTIF
      DO 170 INODE=1,NNODE
      DO 170 IDOFN=1,NDOFN
      NPOSI=(INODE-1)*NDOFN+IDOFN
      LOCNO=LNODS(IELEM,INODE)
      IF(LOCNO.GT.0) LOCEL(NPOSI)=(LOCNO-1)*NDOFN+IDOFN
      IF(LOCNO.LT.0) LOCEL(NPOSI)=(LOCNO+1)*NDOFN-IDOFN
170 CONTINUE
C
C START BY LOOKING FOR EXISTING DESTINATIONS
C
      DO 210 IEVAB=1,NEVAB
      NIKNO=IABS(LOCEL(IEVAB))
      KEXIS=0
      DO 180 IFRON=1,NFRON
      IF(NIKNO.NE.NACVA(IFRON)) GO TO 180
      KEVAB=KEVAB+1
      KEXIS=1
      NDEST(KEVAB)=IFRON
180 CONTINUE
      IF(KEXIS.NE.0) GO TO 210
C
C SEEK NEW EMPTY PLACES FOR DESTINATION FACTOR
C
      DO 190 IFRON=1,MFRON
      IF(NACVA(IFRON).NE.0) GO TO 190
      NACVA(IFRON)=NIKNO
      KEVAB=KEVAB+1
      NDEST(KEVAB)=IFRON
      GO TO 200
190 CONTINUE
C
C THE NEW PLACES MAY DEMAND AN INCREASE IN CURRENT FRONTWIDTH
C
      200 IF(NDEST(KEVAB).GT.NFRON) NFRON=NDEST(KEVAB)
      210 CONTINUE
      WRITE(7) LOCEL,NDEST,NACVA,NFRON
      400 IF(KRESL.GT.1) READ(7) LOCEL,NDEST,NACVA,NFRON
C
C ASSEMBLE ELEMENT LOADS
C
      DO 220 IEVAB=1,NEVAB

```

```

      IDEST=NDEST(IEVAB)
      GLOAD(IDEST)=GLOAD(IDEST)+ELOAD(IELEM,IEVAB)
C
C ASSEMBLE THE ELEMEN STIFFNESSES
C
      IF(KRESL.GT.1) GO TO 402
      DO 222 JEVAB=1,NEVAB
      JDEST=NDEST(JEVAB)
      NGASH=NFUNC(IDEST,JDEST)
      NGISH=NFUNC(JDEST,IDEST)
      IF(JDEST.GE.IDEST) GSTIF(NGASH)=GSTIF(NGASH)+ESTIF(IEVAB,JEVAB)
      IF(JDEST.LT.IDEST) GSTIF(NGISH)=GSTIF(NGISH)+ESTIF(IEVAB,JEVAB)
222  CONTINUE
402  CONTINUE
220  CONTINUE
C
C RE-EXAMINE EACH ELEMENT NODE WHICH CAN BE ELIMINATED
C
      DO 310 IEVAB=1,NEVAB
      NIKNO=-LOC(IEVAB)
      IF(NIKNO.LE.0) GO TO 310
C
C FIND POSITIONS OF VARIABLES READY FOR ELIMINATION
C
      DO 300 IFRON=1,NFRON
      IF(NACVA(IFRON).NE.NIKNO) GO TO 300
      NBUFA=NBUFA+1
C
C WRITE EQUATIONS
C
      IF(NBUFA.LE.MBUFA) GO TO 406
      NBUFA=1
      IF(KRESL.GT.1) GO TO 408
      WRITE(2) EQUAT,EQRHS,NPIVO,NAMEV
      GO TO 406
408  WRITE(4) EQRHS
      READ(2) EQUAT,EQRHS,NPIVO,NAMEV
406  CONTINUE
C
C EXTRACT THE COEFFICIENTS OF THE NEW EQUATION FOR ELIMINATION
C
      IF(KRESL.GT.1) GO TO 404
      DO 230 JFRON=1,MFRON
      IF(IFRON.LT.JFRON) NLOCA=NFUNC(IFRON,JFRON)
      IF(IFRON.GE.JFRON) NLOCA=NFUNC(JFRON,IFRON)
      EQUAT(JFRON,NBUFA)=GSTIF(NLOCA)
230  GSTIF(NLOCA)=0.0
404  CONTINUE
C
      EQRHS(NBUFA)=GLOAD(IFRON)
      GLOAD(IFRON)=0.0
      KELVA=KELVA+1
      NAMEV(NBUFA)=NIKNO
      NPIVO(NBUFA)=IFRON
C
C DEAL WITH PIVOT
C
      PIVOT=EQUAT(IFRON,NBUFA)
      IF(PIVOT.GT.0.0) GO TO 235
      WRITE(6,900) NIKNO,PIVOT
900  FORMAT(1H0,3X,52HNEGATIVE OR ZERO PIVOT ENCOUNTERED FOR VARIABLE N
        .O.,14,10H OF VALUE ,E17.6)
      STOP
235  CONTINUE
      EQUAT(IFRON,NBUFA)=0.0

```



```

C
      IF (IFFIX (NIKNO).EQ.0) GO TO 250
C
C   DEAL WITH A PRESCRIBED DEFLECTION
C
      DO 240 JFRON=1, NFRON
240  GLOAD (JFRON)=GLOAD (JFRON)-FIXED (NIKNO)*EQUAT (JFRON, NBUFA)
      GO TO 280
C
250  DO 270 JFRON=1, NFRON
      GLOAD (JFRON)=GLOAD (JFRON)-EQUAT (JFRON, NBUFA)*EQRHS (NBUFA)/PIVOT
C
      IF (KRESL.GT.1) GO TO 418
      IF (EQUAT (JFRON, NBUFA).EQ.0.0) GO TO 270
      NLOCA=NFUNC (0, JFRON)
      CUREQ=EQUAT (JFRON, NBUFA)
      DO 260 LFRON=1, JFRON
      NGASH=LFRON+NLOCA
260  GSTIF (NGASH)=GSTIF (NGASH)-CUREQ*EQUAT (LFRON, NBUFA)
      /PIVOT
418  CONTINUE
270  CONTINUE
280  EQUAT (IFRON, NBUFA)=PIVOT
C
      NACVA (IFRON)=0
      GO TO 290
C
C   COMPLETE THE ELEMENT LOOP IN THE FORWARD ELIMINATION
C
300  CONTINUE
290  IF (NACVA (NFRON).NE.0) GO TO 310
      NFRON=NFRON-1
      IF (NFRON.GT.0) GO TO 290
310  CONTINUE
320  CONTINUE
      IF (KRESL.EQ.1) WRITE (2) EQUAT, EQRHS, NPIVO, NAMEV
      BACKSPACE 2
C
      DO 340 IELVA=1, KELVA
C
C   READ A NEW BLOCK OF EQUATIONS
C
      IF (NBUFA.NE.0) GO TO 412
      BACKSPACE 2
      READ (2) EQUAT, EQRHS, NPIVO, NAMEV
      BACKSPACE 2
      NBUFA=NBUFA
      IF (KRESL.EQ.1) GO TO 412
      BACKSPACE 4
      READ (4) EQRHS
      BACKSPACE 4
412  CONTINUE
C
C   PREPARE TO BACK SUBSTITUTE FROM THE CURRENT EQUATION
C
      IFRON=NPIVO (NBUFA)
      NIKNO=NAMEV (NBUFA)
      PIVOT=EQUAT (IFRON, NBUFA)
      IF (IFFIX (NIKNO).NE.0) VECRV (IFRON)=FIXED (NIKNO)
      IF (IFFIX (NIKNO).EQ.0) EQUAT (IFRON, NBUFA)=0.0
C
      DO 330 JFRON=1, MFRON
330  EQRHS (NBUFA)=EQRHS (NBUFA)-VECRV (JFRON)*EQUAT (JFRON, NBUFA)
C
C   PUT THE FINAL VALUES

```

```

C
      IF (IFFIX(NIKNO).EQ.0) VECRV(IFRON)=EQRHS(NBUFA)/PIVOT
      IF (IFFIX(NIKNO).NE.0) FIXED(NIKNO)=-EQRHS(NBUFA)
      NBUFA=NBUFA-1
      ASDIS(NIKNO)=VECRV(IFRON)
340  CONTINUE
C
C ADD DISPLACEMENT TO PREVIOUS TOTAL VALUES
C
      DO 345 ITOTV=1,NTOTV
345  TDISP(ITOTV)=TDISP(ITOTV)+ASDIS(ITOTV)
C
      KBOUN=1
      DO 370 IPOIN=1,NPOIN
      NLOCA=(IPOIN-1)*NDOFN
      DO 350 IDOFN=1,NDOFN
      NGUSH=NLOCA+IDOFN
      IF (IFFIX(NGUSH).GT.0) GO TO 360
350  CONTINUE
      GO TO 370
360  DO 510 IDOFN=1,NDOFN
      NGASH=NLOCA+IDOFN
510  TREAC(KBOUN,IDOFN)=TREAC(KBOUN,IDOFN)+FIXED(NGASH)
      KBOUN=KBOUN+1
370  CONTINUE
C
      DO 700 IPOIN=1,NPOIN
      DO 710 IELEM=1,NELEM
      DO 710 INODE=1,NNODE
      NLOCA=IABS(LNODS(IELEM,INODE))
710  IF (IPOIN.EQ.NLOCA) GO TO 720
720  DO 730 IDOFN=1,NDOFN
      NGASH=(INODE-1)*NDOFN+IDOFN
      MGASH=(IPOIN-1)*NDOFN+IDOFN
730  TLOAD(IELEM,NGASH)=TLOAD(IELEM,NGASH)+FIXED(MGASH)
700  CONTINUE
      RETURN
      END

      SUBROUTINE RESTR(ASDIS,EFFST,ELOAD,LNODS,
.          MATNO,MELEM,MMATS,MPOIN,MTOTG,MTOTV,
.          NDOFN,NELEM,NEVAB,NGAUS,NNODE,
.          NPROP,NSTRE,POSGP,PROPS,STRSG,
.          TDISP,WEIGP,EPSTN,KUNLO,AMATX,DMATT,
.          THICK,MLAYR,NLAYR,LARGE)
C
C SUBROUTINE UNTUK MENGEVALUASI YIELD SURFACE DAN
C EQUIVALEN NODAL FORCE
C
      DIMENSION ASDIS(MTOTV),AVECT(5),BMATX(5,45),
.          DMATT(5,5,MMATS),DVECT(5),EFFST(MTOTG),ELDIS(45),
.          ELOAD(MELEM,NEVAB),GVECT(5),LNODS(MELEM,9),
.          MATNO(MELEM,MLAYR),POSGP(5),PROPS(MMATS,NPROP),
.          DESIG(5),SIGMA(5),SGTOT(5),ETDIS(45),
.          STRES(5),EPSTN(MTOTG),TDISP(MTOTV),THICK(MPOIN),
.          STRSG(5,MTOTG),WEIGP(5),
.          AMATX(9,MMATS),GMATX(2,45)
      REWIND 8
      DO 5 IELEM=1,NELEM
      DO 5 IEVAB=1,NEVAB
5  ELOAD(IELEM,IEVAB)=0.0
      KUNLO=0
      KGAUS=0
C
C LOOP OVER EACH ELEMENT

```



```

C
      DO 210 IELEM=1, NELEM
C
C IDENTIFIKASI DISPLASMEN TIAP ELEMEN NODAL POINT
C
      JPOSI=0
      DO 10 INODE=1, NNODE
        LNODE=IABS(LNODS(IELEM, INODE))
        NPOSN=(LNODE-1)*NDOFN
        DO 10 IDOFN=1, NDOFN
          NPOSN=NPOSN+1
          JPOSI=JPOSI+1
          ELDIS(JPOSI)=ASDIS(NPOSN)
          ETDIS(JPOSI)=TDISP(NPOSN)
10    CONTINUE
      KELGS=0
C
      DO 205 IGAUS=1, NGAUS
      DO 205 JGAUS=1, NGAUS
      DO 200 ILAYR=1, NLAYR
        LPROP=MATNO(IELEM, ILAYR)
        UNIAX=PROPS(LPROP, 6)
        HARDS=PROPS(LPROP, 7)
        KGAUS=KGAUS+1
        KELGS=KELGS+1
        EPSTN(KGAUS)=ABS(EPSTN(KGAUS))
        READ(8) BMATX, GMATX, DVOLU
C
C EVALUASI GEOMETRIC NONLINEARITY
C
      IF(LARGE.EQ.1)
        .CALL      LDISP(BMATX, GMATX, ETDIS, NEVAB)
C
      DO 30 IDOFN=1, NDOFN
        GASH=0.0
        DO 25 IEVAB=1, NEVAB
          25 GASH=GASH+BMATX(IDOFN, IEVAB)*ELDIS(IEVAB)
          30 GVECT(IDOFN)=GASH
C
C HITUNG LOCAL STRES
C
      DO 50 ISTRE=1, NSTRE
        GASH=0.0
        DO 45 JSTRE=1, NSTRE
          45 GASH=GASH+DMATT(ISTRE, JSTRE, LPROP)*GVECT(JSTRE)
          50 STRES(ISTRE)=GASH
C
C REDUKSI STRES PADA YIELD SURFACE
C
      PREYS=UNIAX+EPSTN(KGAUS)*HARDS
      DO 150 ISTR1=1, NSTRE
        DESIG(ISTR1)=STRES(ISTR1)
150    SIGMA(ISTR1)=STRSG(ISTR1, KGAUS)+STRES(ISTR1)
        CALL      INVAR(AMATE, SIGMA, LPROP, MMATS, YIELD)
        ESPRE=EFFST(KGAUS)-PREYS
        IF(ESPRE.GE.0.0) GO TO 55
        ESCUR=YIELD-PREYS
        IF(ESCUR.LE.0.0) GO TO 60
        RFACT=ESCUR/(YIELD-EFFST(KGAUS))
        GO TO 70
55    ESCUR=YIELD-EFFST(KGAUS)
        IF(ESCUR.LE.0.0) GO TO 60
        RFACT=1.0
70    MSTEP=ESCUR*8.0/UNIAX+1.0

```

```

      ASTEP=MSTEP
      REDUC=1.0-RFACT
      DO 80 ISTR1=1, NSTRE
        SGTOT(ISTR1)=STRSG(ISTR1, KGAUS)+REDUC*STRES(ISTR1)
      80 STRES(ISTR1)=RFACT*STRES(ISTR1)/ASTEP
      DO 90 ISTEP=1, MSTEP
        CALL      INVAR(AMATX, SGTOT, LPROP, MMATS, YIELD)
        CALL      FLOWS(ABETA, AVECT, DVECT, LPROP,
                      MMATS, NPROP, PROPS, SGTOT, AMATX, DMATT)
        AGASH=0.0
        DO 100 ISTR1=1, NSTRE
      100 AGASH=AGASH+AVECT(ISTR1)*STRES(ISTR1)
          DLAMD=AGASH*ABETA
          IF(DLAMD.LT.0.0) DLAMD=0.0
          BGASH=0.0
          DO 110 ISTR1=1, NSTRE
            BGASH=BGASH+AVECT(ISTR1)*SGTOT(ISTR1)
      110 SGTOT(ISTR1)=SGTOT(ISTR1)+STRES(ISTR1)-DLAMD*DVECT(ISTR1)
          EPSTN(KGAUS)=EPSTN(KGAUS)+DLAMD*BGASH/YIELD
      90 CONTINUE
        CALL      INVAR(AMATX, SGTOT, LPROP, MMATS, YIELD)
        CURYS=UNIAX+EPSTN(KGAUS)*HARDS
        BRING=1.0
        IF(YIELD.GT.CURYS) BRING=CURYS/YIELD
        DO 130 ISTR1=1, NSTRE
      130 STRSG(ISTR1, KGAUS)=BRING*SGTOT(ISTR1)
          EFFST(KGAUS)=BRING*YIELD
C
      GO TO 190
      60 DO 180 ISTR1=1, NSTRE
      180 STRSG(ISTR1, KGAUS)=STRSG(ISTR1, KGAUS)+DESIG(ISTR1)
          EFFST(KGAUS)=YIELD
          IF(EPSTN(KGAUS).EQ.0.0.OR.ESCUR.EQ.0.0) GO TO 190
          EPSTN(KGAUS)=-EPSTN(KGAUS)
          KUNLO=KUNLO+1
      190 CONTINUE
C
C HITUNG EQUIVALENT NODAL FORCE
C
      MGASH=0
      DO 140 INODE=1, NNODE
        DO 140 IDOFN=1, NDOFN
          MGASH=MGASH+1
          DO 140 ISTR1=1, NSTRE
      140 BLOAD(IELEM, MGASH)=BLOAD(IELEM, MGASH)+BMATX(ISTR1, MGASH)*
          .STRSG(ISTR1, KGAUS)*DVOLU
      200 CONTINUE
      205 CONTINUE
      210 CONTINUE
      RETURN
      END

      SUBROUTINE INVAR(A, ST, LPROP, MMATS, YIELD)
C
C SUBROUTINE UNTUK MENGEVALUASI YIELD FUNCTION
C
      DIMENSION ST(5), A(9, MMATS)
      L=LPROP
      GASH=A(1, L)*ST(1)*ST(1)+2.0*A(2, L)*ST(1)*ST(2)+2.0*A(3, L)*
          ST(1)*ST(3)+A(4, L)*ST(2)*ST(2)+2.0*A(5, L)*ST(2)*ST(3)+
          A(6, L)*ST(3)*ST(3)+A(7, L)*ST(4)*ST(4)+2.0*A(8, L)*ST(4)*
          ST(5)+A(9, L)*ST(5)*ST(5)
      YIELD=SQRT(GASH)
      RETURN
      END

```



```

SUBROUTINE CONVER(ELOAD, IITER, LNODS, MELEM, MEVAB, MTOTV, NCHEK,
      NDOFN, NELEM, NEVAB, NNODE, NTOTV, STFOR,
      TLOAD, TOFOR, TOLER)
C
C SUBROUTINE UNTUK MEMERIKSA KONVERGENSI DARI HASIL ITERASI
C
      DIMENSION ELOAD(MELEM, MEVAB), LNODS(MELEM, 12), STFOR(MTOTV),
      TOFOR(MTOTV), TLOAD(MELEM, MEVAB)
      NCHEK=0
      RESID=0.0
      RETOT=0.0
      REMAX=0.0
      DO 5 ITOTV=1, NTOTV
      STFOR(ITOTV)=0.0
      TOFOR(ITOTV)=0.0
5 CONTINUE
      DO 40 IELEM=1, NELEM
      KEVAB=0
      DO 40 INODE=1, NNODE
      LOCNO=IABS(LNODS(IELEM, INODE))
      DO 40 IDOFN=1, NDOFN
      KEVAB=KEVAB+1
      NPOSI=(LOCNO-1)*NDOFN+IDOFN
      STFOR(NPOSI)=STFOR(NPOSI)+ELOAD(IELEM, KEVAB)
40 TOFOR(NPOSI)=TOFOR(NPOSI)+TLOAD(IELEM, KEVAB)
      DO 50 ITOTV=1, NTOTV
      REFOR=TOFOR(ITOTV)-STFOR(ITOTV)
      RESID=RESID+REFOR*REFOR
      RETOT=RETOT+TOFOR(ITOTV)*TOFOR(ITOTV)
      AGASH=ABS(REFOR)
50 IF(AGASH.GT.REMAX) REMAX=AGASH
      DO 10 IELEM=1, NELEM
      DO 10 IEVAB=1, NEVAB
10 ELOAD(IELEM, IEVAB)=TLOAD(IELEM, IEVAB)-ELOAD(IELEM, IEVAB)
      RESID=SQRT(RESID)
      RETOT=SQRT(RETOT)
      RATIO=100.0*RESID/RETOT
      IF(RATIO.GT.TOLER) NCHEK=1
      IF(IITER.EQ.1) GO TO 20
      IF(RATIO.GT.PVALU) NCHEK=999
20 PVALU=RATIO
      RETURN
      END

SUBROUTINE OUTPUT(IITER, MTOTG, MTOTV, MVFIX, NCHEK, NELEM, NGAUS,
      NOFIX, NOUTP, NPOIN, NSTRE,
      NVFIX, STRSG, TDISP, TREAC, EPSTN, POSGP,
      EFFST, MATNO, MMATS, PROPS, NPROP, MELEM, THICK,
      MPOIN, LNODS, MLAYR, NLAYR)
C
C SUBROUTINE UNTUK OUTPUT DISPLASMEN, REAKSI DAN TEGANGAN
C
      DIMENSION NOFIX(MVFIX), NOUTP(2), STRSG(5, MTOTG), STRES(6),
      TDISP(MTOTV), TREAC(MVFIX, 5), EPSTN(MTOTG),
      POSGP(5), SHAPE(3, 9), EFFST(MTOTG),
      THICK(MPOIN), LNODS(MELEM, 9),
      FORCE(8), MATNO(MELEM, MLAYR),
      PROPS(MMATS, NPROP)
      KOUTP=NOUTP(1)
      IF(IITER.GT.1) KOUTP=NOUTP(2)
      IF(IITER.EQ.1.AND.NCHEK.EQ.0) KOUTP=NOUTP(2)
C
C OUTPUT DISPLASMEN
C

```

```

      IF(KOUTP.LT.1) GO TO 10
      WRITE(6,900)
900  FORMAT(1H0,5X,13HDISPLACEMENTS)
      WRITE(6,905)
905  FORMAT(1H0,6X,4HNODE,4X,6HX-DISP,8X,6HY-DISP,8X,6HZ-DISP,
        .      8X,6HAF-ROT,8X,6HBT-ROT)
      DO 20 IPOIN=1,NPOIN
      NGASH=IPOIN*5
      NGISH=NGASH-4
20  WRITE(6,910) IPOIN, (TDISP(IGASH), IGASH=NGISH, NGASH)
910  FORMAT(1I0,5B14.6)
10  CONTINUE

C
C  OUTPUT REAKSI
C
      IF(KOUTP.LT.2) GO TO 30
      WRITE(6,920)
920  FORMAT(1H0,5X,9HREACTIONS)
      WRITE(6,925)
925  FORMAT(1H0,6X,4HNODE,4X,6HX-REAC,8X,6HY-REAC,8X,6HZ-REAC,
        .      8X,6HAF-MOM,8X,6HBT-MOM)
      DO 40 IVFIX=1,NVFIX
40  WRITE(6,910) NOFIX(IVFIX), (TREAC(IVFIX, IDOFN), IDOFN=1,5)
30  CONTINUE

C
C  OUTPUT STRES
C
      IF(KOUTP.LT.3) GO TO 120
      WRITE(6,927)
927  FORMAT(1H0,5X,8HSTRESSES)
      WRITE(6,926)
926  FORMAT(1H0,1X,5HKLAYR,5X,6HXX-STR,8X,6HYY-STR,8X,6HXY-STR,
        .      8X,6HYZ-STR,8X,6HYZ-STR,6X,10HEFF.STRESS,3X,
        .      13HEFF.PL.STRAIN)
      KGAUS=0
      DO 110 IELEM=1, NELEM
      KELGS=0
      WRITE(6,940) IELEM
940  FORMAT(1H0,18H * ELEMENT NO. =,I5,/)
      DO 105 IGAUS=1,NGAUS
      DO 105 JGAUS=1,NGAUS
      EXISP=POSGP(IGAUS)
      ETASP=POSGP(JGAUS)

C
      DO 70 JFORC=1,8
70  FORCE(JFORC)=0.0
      KGASP=0
      KELGS=KELGS+1
      WRITE(6,945) KELGS
945  FORMAT(11H G.P. NO. =,I5)
      CALL      SFR1(SHAPE,EXISP,ETASP)

C
      THIGP=0.0
      DO 65 INODE=1,8
      IPOIN=IABS(LNODS(IELEM, INODE))
65  THIGP=THIGP+SHAPE(1, INODE)*THICK(IPOIN)
      ZETSP=-1.0
      DO 100 ILAYR=1,NLAYR
      LPROP=MATNO(IELEM, ILAYR)
      DZETA=PROPS(LPROP,3)
      ZETSP=ZETSP+DZETA/2.0
      KGAUS=KGAUS+1
      KGASP=KGASP+1

C
C  LOCAL STRESS

```



```

C
DO 50 ISTRE=1,NSTRE
50 STRES(ISTRE)=STRSG(ISTRE,KGAUS)
WRITE(6,950) KGASP,(STRES(ISTRE),ISTRE=1,NSTRE),
EFFST(KGAUS),EPSTN(KGAUS)
C
C SET UP STRESS RESULTANT
C
DO 75 ISTRE=1,3
FORCE(ISTRE)=FORCE(ISTRE)+STRES(ISTRE)*THIGP/2.0*DZETA
75 FORCE(ISTRE+3)=FORCE(ISTRE+3)-STRES(ISTRE)*THIGP*THIGP*
ZETSP*DZETA/4.0
DO 80 ISTRE=4,5
80 FORCE(ISTRE+3)=FORCE(ISTRE+3)+STRES(ISTRE)*THIGP/2.0*DZETA
950 FORMAT(I5,2X,8E14.6)
ZETSP=ZETSP+DZETA/2.0
100 CONTINUE
WRITE(6,960) FORCE(1),FORCE(4),FORCE(2),FORCE(5),FORCE(3),
FORCE(6),FORCE(7),FORCE(8)
960 FORMAT(/,22H STRESS RESULTANTS = ,6HN-XX =,E12.5,5X,
6HM-XX =,E12.5/22X,6HN-YY =,E12.5,5X,6HM-YY =,
E12.5/22X,6HN-XY =,E12.5,5X,6HM-XY =,E12.5/,
22X,6HQ-XZ =,E12.5,5X,6HQ-YZ =,E12.5)
105 CONTINUE
110 CONTINUE
120 CONTINUE
RETURN
END

SUBROUTINE RESTAR(EFFST,ELOAD,EPSTN,MELEM,MEVAB,MTOTG,
MTOTV,MVFIX,TDISP,TLOAD,TREAC,STRSG,
TFACT,KINCS)
C
C SUBROUTINE UNTUK MENYIMPAN DATA YANG AKAN DIGUNAKAN
C PADA PROSES SELANJUTNYA
C
DIMENSION EFFST(MTOTG),ELOAD(MELEM,MEVAB),EPSTN(MTOTG),
TDISP(MTOTV),TLOAD(MELEM,MEVAB),TREAC(MVFIX,5),
STRSG(5,MTOTG)
REWIND 12
WRITE(12) KINCS,TFACT,EFFST,ELOAD,EPSTN
WRITE(12) TDISP,TLOAD,TREAC,STRSG
RETURN
END

```





# **TUGAS AKHIR**

**LAMPIRAN 2**

**LISTING PROGRAM  
PLOT GRAFIS**



## LAMPIRAN 2. LISTING PROGRAM PLOT GRAFIS

```
C234567
      INCLUDE 'FGRAPH.FI'
      INCLUDE 'FGRAPH.FD'
C
C LISTING PROGRAM PLOT GRAFIS
C
      INTEGER*2      error, oldapage, oldvpage
      INTEGER*2 status, xhalf, yhalf, xquar, yquar, stats
      DOUBLE PRECISION x_upleft, y_upleft, x_botright, y_botright
      RECORD / videoconfig / vc
      record/wxycoord /wxy
      DIMENSION COORD(2052,8), XCOORD(2052,8), LNODS(670,8)
      ., DFORM(2052,3),
      .coor1(2052,3), xoor1(2052,3), coor2(2052,3), xoor2(2052,3)
C
      OPEN(1, FILE='C:\PLSHELL\DATA\PLOT.DTA')
C
C BACA DATA KOORDINAT, ELEMEN DAN DISPLASMEN
C
      READ(1,367) NPOIN, NELEM
367 FORMAT(2I5)
      DO 45 IELEM=1, NELEM
      READ(1,900) NUMEL, MATNO, (LNODS(NUMEL, INODE), INODE=1,8)
900 FORMAT(16I5)
      45 CONTINUE
      6 READ(1,905) IPOIN, (COORD(IPOIN, IDIME), IDIME=1,8)
905 FORMAT(I5, 4F15.10/5X, 4F15.10)
      IF(IPOIN.NE.NPOIN) GO TO 6
      DO 17 I=1, NPOIN
      READ(1,978) IPOIN, (DFORM(IPOIN, IGASH), IGASH=1,3)
978 FORMAT(I10, 3E14.6)
      17 CONTINUE
C
      oldapage = getactivepage()
      oldvpage = getvisualpage()
C
      IF( setvideomode( $VRES16COLOR ) .EQ. 0 )
      + STOP 'status: cannot set graphics mode'
C
      WRITE(*, '(7(/))')
      WRITE(*, '(22X,A,\)' ) CHAR(201)
      WRITE(*, '(33A,\)' ) (CHAR(205), I=1,33)
      WRITE(*, '(A)' ) CHAR(187)
      write(*, '(22x,a,2x,29a,a)' ) char(186), ' P L O T G R A F I S
      . , CHAR(186)
      WRITE(*, '(22X,A,\)' ) CHAR(204)
      WRITE(*, '(33A,\)' ) (CHAR(205), I=1,33)
      WRITE(*, '(A)' ) CHAR(185)
      WRITE(*, '(22X,A,\)' ) CHAR(186)
      WRITE(*, '(33X,A)' ) CHAR(186)
      WRITE(*, '(22X,A,2X,29A,A)' ) CHAR(186), 'P R O G R A M A N A L I S
      . A , CHAR(186)
      WRITE(*, '(22X,A,\)' ) CHAR(186)
      WRITE(*, '(33X,A)' ) CHAR(186)
      WRITE(*, '(22X,A,2X,29A,A)' ) CHAR(186), 'E L A S T O - P L A S T I
      . S , CHAR(186)
      WRITE(*, '(22X,A,\)' ) CHAR(186)
      WRITE(*, '(33X,A)' ) CHAR(186)
      WRITE(*, '(22X,A,\)' ) CHAR(200)
      WRITE(*, '(33A,\)' ) (CHAR(205), I=1,33)
      WRITE(*, '(A)' ) CHAR(188)
      WRITE(*, '(7(/))')
```

```

WRITE(*,*)' <CR>'
read(*,*)
CALL CLEARSCREEN($GCLEARSCREEN)
C
587 CONTINUE
write(*,'(////,50A)')
.'          PLOT MODE      : '
write(*,'(//,50A)')
.'          1.UNDEFORMED SHAPE      '
write(*,'(/,50A)')
.'          2.DEFORMED SHAPE        '
write(*,'(/,50A)')
.'          3.UNDEFORMED+DEFORMED SHAPE '
write(*,'(/,50A)')
.'          4.ANIMATION MODE        '
write(*,'(/,50A)')
.'          5.EXIT                  '
C
576 WRITE(*,908)
908 FORMAT(//,14X,'SET PLOT MODE (1-5) :',\\)
READ(*,74)LOT
IF(LOT.GT.5.OR.LOT.LT.1) GOTO 576
IF(LOT.EQ.5)GOTO 705
C
CALL CLEARSCREEN($GCLEARSCREEN)
C
write(*,'(////////,50A)')
.'          VIEW PORT      : '
write(*,'(/,50A)')
.'          1. X-Z          '
write(*,'(50A)')
.'          2. X-Y          '
write(*,'(50A)')
.'          3. Y-Z          '
C
4 WRITE(*,796)
796 FORMAT(//,6X,'SET VIEW-PORT (1-3) :',\\)
READ(*,74)KP
74 FORMAT(I5)
WRITE(*,87)
87 FORMAT(//,6X,'SET SUDUT VIEW-PORT :',\\)
READ(*,174)ALA
174 FORMAT(F5.2)
if(kp.gt.3.or.kp.lt.1)go to 4
C
DO 90 IPOIN=1,NPOIN
COORD(IPOIN,1)=(COORD(IPOIN,1)+DFORM(IPOIN,1))
COORD(IPOIN,2)=(COORD(IPOIN,2)+DFORM(IPOIN,2))
COORD(IPOIN,3)=(COORD(IPOIN,3)+DFORM(IPOIN,3))
90 CONTINUE
C
C set view point
C
i=1
if(kp.eq.1)then
ix=1
iy=3
iz=2
endif
if(kp.eq.2)then
ix=2
iy=1
iz=3
endif

```



```

      if(kp.eq.3)then
      ix=3
      iy=2
      iz=1
      endif
      if(ALA.eq.0.)f=0
C
C SET MAKSIMUM WINDOW
C
      XMAX=XOOR1(1,IX)
      XMIN=XOOR1(1,IX)
      YMAX=XOOR1(1,IY)
      YMIN=XOOR1(1,IY)
      DO 47 IPOIN=1,NPOIN
      XOOR1(IPOIN,ix)=(COORD1(IPOIN,ix)+COORD1(IPOIN,iz)
      . *COS(ALA)*f)*(-1.)
      IF(XOOR1(IPOIN,IX).LT.XMIN)XMIN=XOOR1(IPOIN,IX)
      IF(XOOR1(IPOIN,IX).GT.XMAX)XMAX=XOOR1(IPOIN,IX)
      XOOR1(IPOIN,iy)=(COORD1(IPOIN,iy)+COORD1(IPOIN,iz)
      . *SIN(ALA)*f)*(-1.)
      IF(XOOR1(IPOIN,IY).LT.YMIN)YMIN=XOOR1(IPOIN,IY)
      IF(XOOR1(IPOIN,IY).GT.YMAX)YMAX=XOOR1(IPOIN,IY)
47 CONTINUE
C
      DO 147 IPOIN=1,NPOIN
      xOORD(IPOIN,ix)=(COORD(IPOIN,ix)+COORD(IPOIN,iz)
      . *COS(ala)*f)*(-1.)
      xOORD(IPOIN,iy)=(COORD(IPOIN,iy)+COORD(IPOIN,iz)
      . *SIN(ala)*f)*(-1.)
147 CONTINUE

      IF( setvideomode( $MAXRESMODE ) .EQ. 0 )
      + STOP 'status: cannot set graphics mode'

      CALL getvideoconfig( vc )
      xhalf = vc.numxpixels / 2
      yhalf = vc.numypixels / 2
      xquar = xhalf / 2
      yquar = yhalf / 2
C
      IF(LOT.EQ.1)GOTO 701
      IF(LOT.EQ.2)GOTO 702
      IF(LOT.EQ.3)GOTO 703
      IF(LOT.EQ.4)GOTO 704
C
C PROSES PLOT GRAFIS
C
703 CALL setviewport( 0,0, xhalf*2 , yhalf * 2 )
      status =
      .setwindow( .FALSE.,xmin*1.2,ymin*1.2,xmax*1.2,ymax*1.2 )
      x_upleft = xmin*1.2
      y_upleft = ymin*1.2
      x_botright = xmax*1.2
      y_botright = ymax*1.2
      status = SETCOLOR(1)
      status = rectangle_w( $GFILLINTERIOR,x_upleft, y_upleft,
      + x_botright, y_botright )
      status = setcolor( 7)
      status = rectangle_w( $GBORDER,x_upleft-0.3, y_upleft-0.3,
      + x_botright-0.3, y_botright-0.3 )
      STATUS = SETCOLOR(6)
      DO 55 IELEM=1,NELEM
      DO 56 INODE=1,7,2
      KNODE=INODE+2
      IF(INODE.EQ.7)KNODE=1

```

```

I1=LNODS(IELEM, INODE)
I2=LNODS(IELEM, KNODE)
wx=xOOR1(I1, ix)
wy=xOOR1(I1, iy)
CALL MOVETO_w(wx, wy, wxy)
wx=xOOR1(I2, ix)
wy=xOOR1(I2, iy)
STATUS=LINETO_w (wx, wy)
56 continue
55 continue

STATUS = SETCOLOR(14)
DO 155 IELEM=1, NELEM
DO 156 INODE=1, 7, 2
KNODE=INODE+2
IF (INODE.EQ.7) KNODE=1
I1=LNODS(IELEM, INODE)
I2=LNODS(IELEM, KNODE)
wx=xOOR1(I1, ix)
wy=xOOR1(I1, iy)
CALL MOVETO_w(wx, wy, wxy)
wx=xOOR1(I2, ix)
wy=xOOR1(I2, iy)
STATUS=LINETO_w (wx, wy)
156 continue
155 continue
  READ (*,*) ! Wait for ENTER to be pressed
  CALL CLEARSCREEN($GCLEARSCREEN)
  GOTO 587

702 CALL setviewport( 0, 0, xhalf*2 , yhalf*2 )
  status =
  .setwindow( .FALSE., xmin*1.2, ymin*1.2, xmax*1.2, ymax*1.2 )
  x_upleft = xmin*1.2
  y_upleft = ymin*1.2
  x_botright = xmax*1.2
  y_botright = ymax*1.2
  status = SETCOLOR(1)
  status = rectangle_w( $GFILLINTERIOR, x_upleft, y_upleft,
+ x_botright, y_botright )
  status = setcolor( 7)
  status = rectangle_w( $GBORDER, x_upleft-0.3, y_upleft-0.3,
+ x_botright-0.3, y_botright-0.3 )
  STATUS = SETCOLOR(14)
  DO 25 IELEM=1, NELEM
  DO 26 INODE=1, 7, 2
  KNODE=INODE+2
  IF (INODE.EQ.7) KNODE=1
  I1=LNODS(IELEM, INODE)
  I2=LNODS(IELEM, KNODE)
  wx=xOOR1(I1, ix)
  wy=xOOR1(I1, iy)
  CALL MOVETO_w(wx, wy, wxy)
  wx=xOOR1(I2, ix)
  wy=xOOR1(I2, iy)
  STATUS=LINETO_w (wx, wy)
26 continue
25 continue
  READ (*,*) ! Wait for ENTER to be pressed
  CALL CLEARSCREEN($GCLEARSCREEN)
  GOTO 587

701 CALL setviewport( 0, 0, xhalf*2 , yhalf*2 )
  status =
  .setwindow( .FALSE., xmin*1.2, ymin*1.2, xmax*1.2, ymax*1.2 )

```



```

x_upleft = xmin*1.2
y_upleft = ymin*1.2
x_botright = xmax*1.2
y_botright = ymax*1.2
status = SETCOLOR(1)
status = rectangle_w( $GFILLINTERIOR,x_upleft, y_upleft,
+                      x_botright, y_botright )
status = setcolor( 7)
status = rectangle_w( $GBORDER,x_upleft-0.3, y_upleft-0.3,
+                      x_botright-0.3, y_botright-0.3 )
STATUS = SETCOLOR(14)
DO 125 IELEM=1,NELEM
DO 126 INODE=1,7,2
KNODE=INODE+2
IF (INODE.EQ.7)KNODE=1
I1=LNODS(IELEM,INODE)
I2=LNODS(IELEM,KNODE)
wx=xCOORD(I1,ix)
wy=xCOORD(I1,iy)
CALL MOVETO_w(wx,wy,wxy)
wx=xCOORD(I2,ix)
wy=xCOORD(I2,iy)
STATUS=LINETO_w (wx,wy)
126 continue
125 continue
READ (*,*) ! Wait for ENTER to be pressed
CALL CLEARSCREEN($GCLEARSCREEN)
GOTO 587

704 IF( setvideomode( $mRES16COLOR ) .EQ. 0 )
+ STOP 'Error: cannot set graphics mode'

error=setactivepage(0)
CALL setviewport( 0, 0, xhalf*2 , yhalf*2 )
status =
.setwindow( .FALSE.,xmin*1.2,ymin*1.2,xmax*1.2,ymax*1.2 )
x_upleft = xmin*1.2
y_upleft = ymin*1.2
x_botright = xmax*1.2
y_botright = ymax*1.2
status = SETCOLOR(1)
status = rectangle_w( $GFILLINTERIOR,x_upleft, y_upleft,
+                      x_botright, y_botright )
status = setcolor( 7)
status = rectangle_w( $GBORDER,x_upleft-0.3, y_upleft-0.3,
+                      x_botright-0.3, y_botright-0.3 )
STATUS = SETCOLOR(14)
DO 225 IELEM=1,NELEM
DO 226 INODE=1,7,2
KNODE=INODE+2
IF (INODE.EQ.7)KNODE=1
I1=LNODS(IELEM,INODE)
I2=LNODS(IELEM,KNODE)
wx=xCOORD(I1,ix)
wy=xCOORD(I1,iy)
CALL MOVETO_w(wx,wy,wxy)
wx=xCOORD(I2,ix)
wy=xCOORD(I2,iy)
STATUS=LINETO_w (wx,wy)
226 continue
225 continue
C
C PLOT ANIMASI MODE
C
do i=1,4

```

```

error=SETACTIVEPAGE(i)

DO 904 IPOIN=1,NPOIN
  COOR2(IPOIN,1)=(COORD(IPOIN,1)+DFORM(IPOIN,1)/4*i)
  COOR2(IPOIN,2)=(COORD(IPOIN,2)+DFORM(IPOIN,2)/4*i)
  COOR2(IPOIN,3)=(COORD(IPOIN,3)+DFORM(IPOIN,3)/4*i)
904 CONTINUE

DO 143 IPOIN=1,NPOIN
  xOOR2(IPOIN,ix)=(COOR2(IPOIN,ix)+COOR2(IPOIN,iz)
    . *COS(ala)*f)*(-1.)
  xOOR2(IPOIN,iy)=(COOR2(IPOIN,iy)+COOR2(IPOIN,iz)
    . *SIN(ala)*f)*(-1.)
143 CONTINUE

CALL setviewport( 0, 0, xhalf*2 , yhalf*2 )
status =
.setwindow( .FALSE.,xmin*1.2,ymin*1.2,xmax*1.2,ymax*1.2 )
x_upleft = xmin*1.2
y_upleft = ymin*1.2
x_botright = xmax*1.2
y_botright = ymax*1.2
status = SETCOLOR(1)
status = rectangle_w( $GFILLINTERIOR,x_upleft, y_upleft,
+ x_botright, y_botright )
status = setcolor( 7)
status = rectangle_w( $GBORDER,x_upleft-0.3, y_upleft-0.3,
+ x_botright-0.3, y_botright-0.3 )
STATUS = SETCOLOR(14)
DO 259 IELEM=1,NELEM
DO 269 INODE=1,7,2
  KNODE=INODE+2
  IF(INODE.EQ.7)KNODE=1
  I1=LNODS(IELEM,INODE)
  I2=LNODS(IELEM,KNODE)
  wx=xOOR2(I1,ix)
  wy=xOOR2(I1,iy)
  CALL MOVETO_w(wx,wy,wxy)
  wx=xOOR2(I2,ix)
  wy=xOOR2(I2,iy)
  STATUS=LINETO_w (wx,wy)
269 continue
259 continue
      end do

  ik=1
  do i=5,7
    error=SETACTIVEPAGE(i)
    ik=ik+2
    if=i-ik
    DO 903 IPOIN=1,NPOIN
      COOR2(IPOIN,1)=(COORD(IPOIN,1)+DFORM(IPOIN,1)/4*if)
      COOR2(IPOIN,2)=(COORD(IPOIN,2)+DFORM(IPOIN,2)/4*if)
      COOR2(IPOIN,3)=(COORD(IPOIN,3)+DFORM(IPOIN,3)/4*if)
903 CONTINUE

    DO 353 IPOIN=1,NPOIN
      xOOR2(IPOIN,ix)=(COOR2(IPOIN,ix)+COOR2(IPOIN,iz)
        . *COS(ala)*f)*(-1.)
      xOOR2(IPOIN,iy)=(COOR2(IPOIN,iy)+COOR2(IPOIN,iz)
        . *SIN(ala)*f)*(-1.)
353 CONTINUE

    CALL setviewport( 0, 0, xhalf*2 , yhalf*2 )
    status =

```



```

.setwindow( .FALSE., xmin*1.2, ymin*1.2, xmax*1.2, ymax*1.2 )
x_upleft = xmin*1.2
y_upleft = ymin*1.2
x_botright = xmax*1.2
y_botright = ymax*1.2
status = SETCOLOR(1)
status = rectangle_w( $GFILLINTERIOR, x_upleft, y_upleft,
+                      x_botright, y_botright )
status = setcolor( 7 )
status = rectangle_w( $GBORDER, x_upleft-0.3, y_upleft-0.3,
+                      x_botright-0.3, y_botright-0.3 )
STATUS = SETCOLOR(14)
DO 219 IELEM=1, NELEM
DO 319 INODE=1, 7, 2
KNODE=INODE+2
IF (INODE.EQ.7) KNODE=1
I1=LNODS(IELEM, INODE)
I2=LNODS(IELEM, KNODE)
wx=xOOR2(I1, ix)
wy=xOOR2(I1, iy)
CALL MOVETO_w(wx, wy, wxy)
wx=xOOR2(I2, ix)
wy=xOOR2(I2, iy)
STATUS=LINETO_w (wx, wy)
319 continue
219 continue
      end do

do j=1,20
DO im = 0, 7
      error=setvisualpage(im)
      call delay(int2(5))
END DO
END DO
READ (*,*) ! Wait for ENTER to be pressed
CALL CLEARSCREEN($GCLEARSCREEN)
status=setvideomode($maxresmode)
GOTO 587

705 status = setvideomode( $DEFAULTMODE )
END

CC DELAY - Pauses for a specified number of 1/100 seconds.
CC
CC Params: Wait - pause time in 0.01 seconds

SUBROUTINE delay( wait )
INTEGER*2 wait, tick0, tick1, tick, kount, dummy

kount = 0
CALL GETTIM( dummy, dummy, dummy, tick0 )
DO WHILE( kount .LT. wait )
  CALL GETTIM( dummy, dummy, dummy, tick1 )
  tick = tick1 - tick0
  IF( tick .LT. 0 ) tick = tick + 100
  tick0 = tick1
  kount = kount + tick
END DO
RETURN
END

```





# **TUGAS AKHIR**

## **LAMPIRAN 3**

### **CONTOH INPUT DATA PROGRAM**



### LAMPIRAN 3. CONTOH INPUT DATA PROGRAM

PLASTO SHELL

692	214	96	8	1	2	2	0	2	1	1	1	1
0.0		0.0		0.0		0.0						
1	1	1	18	27	28	29	19	3	2			
2	1	3	19	29	30	31	20	5	4			
3	1	5	20	31	32	33	21	7	6			
4	1	7	21	33	34	35	22	9	8			
5	1	9	22	35	36	37	23	11	10			
6	1	11	23	37	38	39	24	13	12			
7	1	13	24	39	40	41	25	15	14			
8	1	15	25	41	42	43	26	17	16			
9	1	17	26	42	43	44	27	19	18			
10	1	19	27	44	45	46	28	21	20			
11	1	21	28	45	46	47	29	23	22			
12	1	23	29	46	47	48	30	25	24			
13	1	25	30	47	48	49	31	27	26			
14	1	27	31	48	49	50	32	29	28			
15	1	29	32	49	50	51	33	31	30			
16	1	31	33	50	51	52	34	33	32			
17	1	33	34	51	52	53	35	35	34			
18	1	35	35	52	53	54	36	37	36			
19	1	37	36	53	54	55	37	39	38			
20	1	39	37	54	55	56	38	41	40			
21	1	41	38	55	56	57	39	43	42			
22	1	43	39	56	57	58	40	45	44			
23	1	45	40	57	58	59	41	47	46			
24	1	47	41	58	59	60	42	49	48			
25	1	49	42	59	60	61	43	51	50			
26	1	51	43	60	61	62	44	53	52			
27	1	53	44	61	62	63	45	55	54			
28	1	55	45	62	63	64	46	57	56			
29	1	57	46	63	64	65	47	59	58			
30	1	59	47	64	65	66	48	61	60			
31	1	61	48	65	66	67	49	63	62			
32	1	63	49	66	67	68	50	65	64			
33	1	65	50	67	68	69	51	67	66			
34	1	67	51	68	69	70	52	69	68			
35	1	69	52	69	70	71	53	71	70			
36	1	71	53	70	71	72	54	73	72			
37	1	73	54	71	72	73	55	75	74			
38	1	75	55	72	73	74	56	77	76			
39	1	77	56	73	74	75	57	79	78			
40	1	79	57	74	75	76	58	81	80			
41	1	81	58	75	76	77	59	83	82			
42	1	83	59	76	77	78	60	85	84			
43	1	85	60	77	78	79	61	87	86			
44	1	87	61	78	79	80	62	89	88			
45	1	89	62	79	80	81	63	91	90			
46	1	91	63	80	81	82	64	93	92			
47	1	93	64	81	82	83	65	95	94			
48	1	95	65	82	83	84	66	97	96			
49	1	97	66	83	84	85	67	99	98			
50	1	99	67	84	85	86	68	101	100			
...												
200	1	639	649	549	565	575	650	641	640			
201	1	633	646	471	472	473	662	652	651			
202	1	652	662	473	474	475	663	654	653			
203	1	654	663	475	476	477	664	656	655			
204	1	656	664	477	478	479	665	658	657			
205	1	658	665	479	480	481	666	660	659			
206	1	660	666	481	482	483	668	675	661			



207	1	667	684	379	389	405	685	669	668		
208	1	669	685	405	415	431	686	671	670		
209	1	671	686	431	441	457	687	673	672		
210	1	673	687	457	467	483	688	675	674		
211	1	675	688	483	493	509	689	677	676		
212	1	677	689	509	519	535	690	679	678		
213	1	679	690	535	545	561	691	681	680		
214	1	681	691	561	571	587	692	683	682		
1		82.6000000000	-200.0000000000				.0000000000			.0000000000	
		77.4000000000	-200.0000000000				.0000000000			.0000000000	
2		82.6000000000	-50.0000000000				.0000000000			.0000000000	
		77.4000000000	-50.0000000000				.0000000000			.0000000000	
3		82.6000000000	100.0000000000				.0000000000			.0000000000	
		77.4000000000	100.0000000000				.0000000000			.0000000000	
4		82.6000000000	127.5300000000				.0000000000			.0000000000	
		77.4000000000	127.5300000000				.0000000000			.0000000000	
5		82.6000000000	155.0600000000				.0000000000			.0000000000	
		77.4000000000	155.0600000000				.0000000000			.0000000000	
6		82.6000000000	182.5900000000				.0000000000			.0000000000	
		77.4000000000	182.5900000000				.0000000000			.0000000000	
7		82.6000000000	210.1200000000				.0000000000			.0000000000	
		77.4000000000	210.1200000000				.0000000000			.0000000000	
8		82.6000000000	237.6500000000				.0000000000			.0000000000	
		77.4000000000	237.6500000000				.0000000000			.0000000000	
9		82.6000000000	265.1800000000				.0000000000			.0000000000	
		77.4000000000	265.1800000000				.0000000000			.0000000000	
10		82.6000000000	292.7100000000				.0000000000			.0000000000	
		77.4000000000	292.7100000000				.0000000000			.0000000000	
11		82.6000000000	320.2400000000				.0000000000			.0000000000	
		77.4000000000	320.2400000000				.0000000000			.0000000000	
12		82.6000000000	347.7700000000				.0000000000			.0000000000	
		77.4000000000	347.7700000000				.0000000000			.0000000000	
13		82.6000000000	375.3000000000				.0000000000			.0000000000	
		77.4000000000	375.3000000000				.0000000000			.0000000000	
14		82.6000000000	402.8300000000				.0000000000			.0000000000	
		77.4000000000	402.8300000000				.0000000000			.0000000000	
15		82.6000000000	430.3600000000				.0000000000			.0000000000	
		77.4000000000	430.3600000000				.0000000000			.0000000000	
16		82.6000000000	580.3600000000				.0000000000			.0000000000	
		77.4000000000	580.3600000000				.0000000000			.0000000000	
17		82.6000000000	730.3600000000				.0000000000			.0000000000	
		77.4000000000	730.3600000000				.0000000000			.0000000000	
18		81.8933500000	-200.0000000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	-200.0000000000	-10.1027300000			.0000000000			.0000000000	
19		81.8933500000	100.0000000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	100.0000000000	-10.1027300000			.0000000000			.0000000000	
20		81.8933500000	155.0600000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	155.0600000000	-10.1027300000			.0000000000			.0000000000	
21		81.8933500000	210.1200000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	210.1200000000	-10.1027300000			.0000000000			.0000000000	
22		81.8933500000	265.1800000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	265.1800000000	-10.1027300000			.0000000000			.0000000000	
23		81.8933500000	320.2400000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	320.2400000000	-10.1027300000			.0000000000			.0000000000	
24		81.8933500000	375.3000000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	375.3000000000	-10.1027300000			.0000000000			.0000000000	
25		81.8933500000	580.3600000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	580.3600000000	-10.1027300000			.0000000000			.0000000000	
26		81.8933500000	730.3600000000	-10.7814600000			.0000000000			.0000000000	
		76.7378400000	730.3600000000	-10.1027300000			.0000000000			.0000000000	
27		79.7854700000	-200.0000000000	-21.3784500000			.0000000000			.0000000000	
		74.7626600000	-200.0000000000	-20.0325900000			.0000000000			.0000000000	
28		79.7854700000	-50.0000000000	-21.3784500000			.0000000000			.0000000000	
		74.7626600000	-50.0000000000	-20.0325900000			.0000000000			.0000000000	
29		79.7854700000	100.0000000000	-21.3784500000			.0000000000			.0000000000	



	74.7626600000	100.0000000000	-20.0325900000	.0000000000
30	79.7854700000	127.5300000000	-21.3784500000	.0000000000
	74.7626600000	127.5300000000	-20.0325900000	.0000000000
31	79.7854700000	155.0600000000	-21.3784500000	.0000000000
	74.7626600000	155.0600000000	-20.0325900000	.0000000000
32	79.7854700000	182.5900000000	-21.3784500000	.0000000000
	74.7626600000	182.5900000000	-20.0325900000	.0000000000
33	79.7854700000	210.1200000000	-21.3784500000	.0000000000
	74.7626600000	210.1200000000	-20.0325900000	.0000000000
34	79.7854700000	237.6500000000	-21.3784500000	.0000000000
	74.7626600000	237.6500000000	-20.0325900000	.0000000000
35	79.7854700000	265.1800000000	-21.3784500000	.0000000000
	74.7626600000	265.1800000000	-20.0325900000	.0000000000
36	79.7854700000	292.7100000000	-21.3784500000	.0000000000
	74.7626600000	292.7100000000	-20.0325900000	.0000000000
37	79.7854700000	320.2400000000	-21.3784500000	.0000000000
	74.7626600000	320.2400000000	-20.0325900000	.0000000000
38	79.7854700000	347.7700000000	-21.3784500000	.0000000000
	74.7626600000	347.7700000000	-20.0325900000	.0000000000
39	79.7854700000	375.3000000000	-21.3784500000	.0000000000
	74.7626600000	375.3000000000	-20.0325900000	.0000000000
40	79.7854700000	402.8300000000	-21.3784500000	.0000000000
	74.7626600000	402.8300000000	-20.0325900000	.0000000000
41	79.7854700000	430.3600000000	-21.3784500000	.0000000000
	74.7626600000	430.3600000000	-20.0325900000	.0000000000
42	79.7854700000	580.3600000000	-21.3784500000	.0000000000
	74.7626600000	580.3600000000	-20.0325900000	.0000000000
43	79.7854700000	730.3600000000	-21.3784500000	.0000000000
	74.7626600000	730.3600000000	-20.0325900000	.0000000000
44	76.3124500000	-200.0000000000	-31.6096500000	.0000000000
	71.5082800000	-200.0000000000	-29.6197000000	.0000000000
45	76.3124500000	100.0000000000	-31.6096500000	.0000000000
	71.5082800000	100.0000000000	-29.6197000000	.0000000000
46	76.3124500000	155.0600000000	-31.6096500000	.0000000000
	71.5082800000	155.0600000000	-29.6197000000	.0000000000
47	76.3124500000	210.1200000000	-31.6096500000	.0000000000
	71.5082800000	210.1200000000	-29.6197000000	.0000000000
48	76.3124500000	265.1800000000	-31.6096500000	.0000000000
	71.5082800000	265.1800000000	-29.6197000000	.0000000000
49	76.3124500000	320.2400000000	-31.6096500000	.0000000000
	71.5082800000	320.2400000000	-29.6197000000	.0000000000
50	76.3124500000	375.3000000000	-31.6096500000	.0000000000
	71.5082800000	375.3000000000	-29.6197000000	.0000000000
....				
682	59.9050200000	425.8600000000	382.6000000000	.0000000000
	59.9050200000	434.8600000000	382.6000000000	.0000000000
683	66.0801700000	425.8600000000	382.6000000000	.0000000000
	66.0801700000	434.8600000000	382.6000000000	.0000000000
684	-66.0801700000	425.8600000000	232.6000000000	.0000000000
	-66.0801700000	434.8600000000	232.6000000000	.0000000000
685	-52.9259000000	425.8600000000	232.6000000000	.0000000000
	-52.9259000000	434.8600000000	232.6000000000	.0000000000
686	-36.9399700000	425.8600000000	232.6000000000	.0000000000
	-36.9399700000	434.8600000000	232.6000000000	.0000000000
687	-18.9776600000	425.8600000000	232.6000000000	.0000000000
	-18.9776600000	434.8600000000	232.6000000000	.0000000000
688	.0000000000	425.8600000000	232.6000000000	.0000000000
	.0000000000	434.8600000000	232.6000000000	.0000000000
689	18.9776600000	425.8600000000	232.6000000000	.0000000000
	18.9776600000	434.8600000000	232.6000000000	.0000000000
690	36.9399700000	425.8600000000	232.6000000000	.0000000000
	36.9399700000	434.8600000000	232.6000000000	.0000000000
691	52.9259000000	425.8600000000	232.6000000000	.0000000000
	52.9259000000	434.8600000000	232.6000000000	.0000000000

	66.0801700000	425.8600000000	232.6000000000	.0000000000
692	66.0801700000	434.8600000000	232.6000000000	.0000000000
1	10000	0	0	0
17	10000	0	0	0
18	10100	0	0	0
26	10100	0	0	0
27	10100	0	0	0
43	10100	0	0	0
44	10100	0	0	0
52	10100	0	0	0
53	10100	0	0	0
69	10100	0	0	0
70	10100	0	0	0
78	10100	0	0	0
79	10100	0	0	0
95	10100	0	0	0
96	10100	0	0	0
104	10100	0	0	0
105	10100	0	0	0
121	10100	0	0	0
122	10100	0	0	0
130	10100	0	0	0
131	10100	0	0	0
147	10100	0	0	0
148	10100	0	0	0
156	10100	0	0	0
157	00100	0	0	0
173	00100	0	0	0
174	10100	0	0	0
182	10100	0	0	0
183	10100	0	0	0
199	10100	0	0	0
200	10100	0	0	0
208	10100	0	0	0
209	10100	0	0	0
225	10100	0	0	0
226	10100	0	0	0
234	10100	0	0	0
235	10100	0	0	0
251	10100	0	0	0
252	10100	0	0	0
260	10100	0	0	0
261	10100	0	0	0
277	10100	0	0	0
278	10100	0	0	0
286	10100	0	0	0
287	10100	0	0	0
303	10100	0	0	0
304	10100	0	0	0
312	10100	0	0	0
313	10000	0	0	0
329	10000	0	0	0
330	10100	0	0	0
338	10100	0	0	0
339	10100	0	0	0
355	10100	0	0	0
356	10100	0	0	0
364	10100	0	0	0
365	10100	0	0	0
381	10100	0	0	0
382	10100	0	0	0
390	10100	0	0	0
391	10100	0	0	0
407	10100	0	0	0
408	10100	0	0	0









# **TUGAS AKHIR**

## **LAMPIRAN 4**

**CONTOH OUTPUT DATA  
KONTROL ELEMEN  
DAN DISPLASMEN**



# LAMPIRAN 4. CONTOH OUTPUT DATA KONTROL ELEMEN DAN DISPLASMEN

PLASTO SHELL

NPOIN = 692  
 NELEM = 214  
 NVFIX = 96  
 NNODE = 8  
 NMATS = 1  
 NGAUS = 2  
 NGAUZ = 2  
 NEVAB = 40  
 NCOLA = 0  
 NALGO = 2  
 NINCS = 1  
 NLAYR = 1  
 LARGE = 1  
 NREST = 1

X-GRAVITY Y-GRAVITY Z-GRAVITY ANG VEL  
 .00000 .00000 .00000 .00000

ELEMENT	PROPERTY/LAYER	NODE NUMBERS							
1	1	1	18	27	28	29	19	3	2
2	1	3	19	29	30	31	20	5	4
3	1	5	20	31	32	33	21	7	6
4	1	7	21	33	34	35	22	9	8
5	1	9	22	35	36	37	23	11	10
6	1	11	23	37	38	39	24	13	12
7	1	13	24	39	40	41	25	15	14
8	1	15	25	41	42	43	26	17	16
9	1	27	44	53	54	55	45	29	28
10	1	29	45	55	56	57	46	31	30

11	1	31	46	57	58	59	47	33	32
12	1	33	47	59	60	61	48	35	34
13	1	35	48	61	62	63	49	37	36
14	1	37	49	63	64	65	50	39	38
15	1	39	50	65	66	67	51	41	40
16	1	41	51	67	68	69	52	43	42
17	1	53	70	79	80	81	71	55	54
18	1	55	71	81	82	83	72	57	56
19	1	57	72	83	84	85	73	59	58
20	1	59	73	85	86	87	74	61	60

....

200	1	646	656	473	474	475	657	648	647
201	1	648	657	475	476	477	658	650	649
202	1	650	658	477	478	479	659	652	651
203	1	652	659	479	480	481	660	654	653
204	1	654	660	481	482	483	677	667	655
205	1	661	674	405	415	431	675	663	662
206	1	663	675	431	441	457	676	665	664
207	1	665	676	457	467	483	677	667	666
208	1	667	677	483	493	509	678	669	668
209	1	669	678	509	519	535	679	671	670
210	1	671	679	535	545	561	680	673	672
211	1	675	688	483	493	509	689	677	676
212	1	677	689	509	519	535	690	679	678
213	1	679	690	535	545	561	691	681	680
214	1	681	691	561	571	587	692	683	682

NODE	X	Y	Z	PRESS
1	82.6000000000-200.0000000000		.0000000000	.0000000000
	77.4000000000-200.0000000000		.0000000000	.0000000000
2	82.6000000000 -50.0000000000		.0000000000	.0000000000
	77.4000000000 -50.0000000000		.0000000000	.0000000000
3	82.6000000000 100.0000000000		.0000000000	.0000000000
	77.4000000000 100.0000000000		.0000000000	.0000000000
4	82.6000000000 127.5300000000		.0000000000	.0000000000
	77.4000000000 127.5300000000		.0000000000	.0000000000



5	82.6000000000	155.0600000000	.0000000000	.0000000000
	77.4000000000	155.0600000000	.0000000000	.0000000000
6	82.6000000000	182.5900000000	.0000000000	.0000000000
	77.4000000000	182.5900000000	.0000000000	.0000000000
7	82.6000000000	210.1200000000	.0000000000	.0000000000
	77.4000000000	210.1200000000	.0000000000	.0000000000
8	82.6000000000	237.6500000000	.0000000000	.0000000000
	77.4000000000	237.6500000000	.0000000000	.0000000000
9	82.6000000000	265.1800000000	.0000000000	.0000000000
	77.4000000000	265.1800000000	.0000000000	.0000000000
10	82.6000000000	292.7100000000	.0000000000	.0000000000
	77.4000000000	292.7100000000	.0000000000	.0000000000
11	82.6000000000	320.2400000000	.0000000000	.0000000000
	77.4000000000	320.2400000000	.0000000000	.0000000000
12	82.6000000000	347.7700000000	.0000000000	.0000000000
	77.4000000000	347.7700000000	.0000000000	.0000000000
13	82.6000000000	375.3000000000	.0000000000	.0000000000
	77.4000000000	375.3000000000	.0000000000	.0000000000
14	82.6000000000	402.8300000000	.0000000000	.0000000000
	77.4000000000	402.8300000000	.0000000000	.0000000000
15	82.6000000000	430.3600000000	.0000000000	.0000000000
	77.4000000000	430.3600000000	.0000000000	.0000000000
16	82.6000000000	580.3600000000	.0000000000	.0000000000
	77.4000000000	580.3600000000	.0000000000	.0000000000
17	82.6000000000	730.3600000000	.0000000000	.0000000000
	77.4000000000	730.3600000000	.0000000000	.0000000000
18	81.8933500000	-200.0000000000	-10.7814600000	.0000000000
	76.7378400000	-200.0000000000	-10.1027300000	.0000000000
19	81.8933500000	100.0000000000	-10.7814600000	.0000000000
	76.7378400000	100.0000000000	-10.1027300000	.0000000000
20	81.8933500000	155.0600000000	-10.7814600000	.0000000000
	76.7378400000	155.0600000000	-10.1027300000	.0000000000
...				
680	45.2364900000	425.8600000000	382.6000000000	.0000000000
	45.3886700000	434.8600000000	382.6000000000	.0000000000
681	52.9259000000	425.8600000000	382.6000000000	.0000000000
	52.9259000000	434.8600000000	382.6000000000	.0000000000

682	59.9050200000	425.8600000000	382.6000000000	.0000000000
	59.9050200000	434.8600000000	382.6000000000	.0000000000
683	66.0801700000	425.8600000000	382.6000000000	.0000000000
	66.0801700000	434.8600000000	382.6000000000	.0000000000
684	-66.0801700000	425.8600000000	232.6000000000	.0000000000
	-66.0801700000	434.8600000000	232.6000000000	.0000000000
685	-52.9259000000	425.8600000000	232.6000000000	.0000000000
	-52.9259000000	434.8600000000	232.6000000000	.0000000000
686	-36.9399700000	425.8600000000	232.6000000000	.0000000000
	-36.9399700000	434.8600000000	232.6000000000	.0000000000
687	-18.9776600000	425.8600000000	232.6000000000	.0000000000
	-18.9776600000	434.8600000000	232.6000000000	.0000000000
688	.0000000000	425.8600000000	232.6000000000	.0000000000
	.0000000000	434.8600000000	232.6000000000	.0000000000
689	18.9776600000	425.8600000000	232.6000000000	.0000000000
	18.9776600000	434.8600000000	232.6000000000	.0000000000
690	36.9399700000	425.8600000000	232.6000000000	.0000000000
	36.9399700000	434.8600000000	232.6000000000	.0000000000
691	52.9259000000	425.8600000000	232.6000000000	.0000000000
	52.9259000000	434.8600000000	232.6000000000	.0000000000
692	66.0801700000	425.8600000000	232.6000000000	.0000000000
	66.0801700000	434.8600000000	232.6000000000	.0000000000

NODE	CODE	FIXED VALUES				
1	10000	.000000	.000000	.000000	.000000	.000000
17	10000	.000000	.000000	.000000	.000000	.000000
18	10100	.000000	.000000	.000000	.000000	.000000
26	10100	.000000	.000000	.000000	.000000	.000000
27	10100	.000000	.000000	.000000	.000000	.000000
43	10100	.000000	.000000	.000000	.000000	.000000
44	10100	.000000	.000000	.000000	.000000	.000000
52	10100	.000000	.000000	.000000	.000000	.000000
53	10100	.000000	.000000	.000000	.000000	.000000
69	10100	.000000	.000000	.000000	.000000	.000000
70	10100	.000000	.000000	.000000	.000000	.000000
78	10100	.000000	.000000	.000000	.000000	.000000
79	10100	.000000	.000000	.000000	.000000	.000000



95	10100	.000000	.000000	.000000	.000000
96	10100	.000000	.000000	.000000	.000000
104	10100	.000000	.000000	.000000	.000000
105	10100	.000000	.000000	.000000	.000000
121	10100	.000000	.000000	.000000	.000000
122	10100	.000000	.000000	.000000	.000000
130	10100	.000000	.000000	.000000	.000000
131	10100	.000000	.000000	.000000	.000000
147	10100	.000000	.000000	.000000	.000000
148	10100	.000000	.000000	.000000	.000000
156	10100	.000000	.000000	.000000	.000000
157	100	.000000	.000000	.000000	.000000
173	100	.000000	.000000	.000000	.000000
174	10100	.000000	.000000	.000000	.000000
182	10100	.000000	.000000	.000000	.000000
183	10100	.000000	.000000	.000000	.000000
199	10100	.000000	.000000	.000000	.000000
200	10100	.000000	.000000	.000000	.000000
208	10100	.000000	.000000	.000000	.000000
209	10100	.000000	.000000	.000000	.000000
225	10100	.000000	.000000	.000000	.000000
226	10100	.000000	.000000	.000000	.000000
234	10100	.000000	.000000	.000000	.000000
235	10100	.000000	.000000	.000000	.000000
251	10100	.000000	.000000	.000000	.000000
252	10100	.000000	.000000	.000000	.000000
260	10100	.000000	.000000	.000000	.000000
261	10100	.000000	.000000	.000000	.000000
277	10100	.000000	.000000	.000000	.000000
278	10100	.000000	.000000	.000000	.000000
286	10100	.000000	.000000	.000000	.000000
287	10100	.000000	.000000	.000000	.000000
303	10100	.000000	.000000	.000000	.000000
304	10100	.000000	.000000	.000000	.000000
312	10100	.000000	.000000	.000000	.000000
313	10000	.000000	.000000	.000000	.000000
329	10000	.000000	.000000	.000000	.000000
330	10100	.000000	.000000	.000000	.000000

334	10100	.000000	.000000	.000000	.000000
339	10100	.000000	.000000	.000000	.000000
355	10100	.000000	.000000	.000000	.000000
356	10100	.000000	.000000	.000000	.000000
364	10100	.000000	.000000	.000000	.000000
365	10100	.000000	.000000	.000000	.000000
381	10100	.000000	.000000	.000000	.000000
382	10100	.000000	.000000	.000000	.000000
390	10100	.000000	.000000	.000000	.000000
391	10100	.000000	.000000	.000000	.000000
407	10100	.000000	.000000	.000000	.000000
408	10100	.000000	.000000	.000000	.000000
416	10100	.000000	.000000	.000000	.000000
417	10100	.000000	.000000	.000000	.000000
433	10100	.000000	.000000	.000000	.000000
434	10100	.000000	.000000	.000000	.000000
442	10100	.000000	.000000	.000000	.000000
443	10100	.000000	.000000	.000000	.000000
459	10100	.000000	.000000	.000000	.000000
460	10100	.000000	.000000	.000000	.000000
468	10100	.000000	.000000	.000000	.000000
469	100	.000000	.000000	.000000	.000000
485	100	.000000	.000000	.000000	.000000
486	10100	.000000	.000000	.000000	.000000
494	10100	.000000	.000000	.000000	.000000
495	10100	.000000	.000000	.000000	.000000
511	10100	.000000	.000000	.000000	.000000
512	10100	.000000	.000000	.000000	.000000
520	10100	.000000	.000000	.000000	.000000
521	10100	.000000	.000000	.000000	.000000
537	10100	.000000	.000000	.000000	.000000
538	10100	.000000	.000000	.000000	.000000
546	10100	.000000	.000000	.000000	.000000
547	10100	.000000	.000000	.000000	.000000
563	10100	.000000	.000000	.000000	.000000
564	10100	.000000	.000000	.000000	.000000
572	10100	.000000	.000000	.000000	.000000
573	10100	.000000	.000000	.000000	.000000



589	10100	.000000	.000000	.000000	.000000	.000000
590	10100	.000000	.000000	.000000	.000000	.000000
598	10100	.000000	.000000	.000000	.000000	.000000
599	10100	.000000	.000000	.000000	.000000	.000000
615	10100	.000000	.000000	.000000	.000000	.000000
616	10100	.000000	.000000	.000000	.000000	.000000
624	10100	.000000	.000000	.000000	.000000	.000000

NUMAT	ELEMENT PROPERTIES						
1	.20000E+06	.30000E+00	.20000E-04	.00000E+00	.00000E+00	.30800E+03	.10000E+03
	.20000E+06	.84000E+05	.84000E+05	.84000E+05	.30800E+03	.30800E+03	.18518E+03
	.18518E+03	.18518E+03	.00000E+00				

MAXIMUM FRONTWIDTH ENCOUNTERED = 415

COEFF(1) = .83333330E+00      COEFF(2) = .83333330E+00

NELEM =	1	NUCLO =	0
NELEM =	2	NUCLO =	0
NELEM =	3	NUCLO =	0
NELEM =	4	NUCLO =	0
NELEM =	5	NUCLO =	0
NELEM =	6	NUCLO =	0
NELEM =	7	NUCLO =	0
NELEM =	8	NUCLO =	0
NELEM =	9	NUCLO =	0
NELEM =	10	NUCLO =	0
...			

NELEM =	185	NUCLO =	0
NELEM =	186	NUCLO =	0
NELEM =	187	NUCLO =	0
NELEM =	188	NUCLO =	0
NELEM =	189	NUCLO =	0

NELEM =	190	NUCLO =	0
NELEM =	191	NUCLO =	0
NELEM =	192	NUCLO =	0
NELEM =	193	NUCLO =	3
LPOIN =	1	LDOFN =	3 LOAD = -1.80890000
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000
NELEM =	194	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000
NELEM =	195	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000
NELEM =	196	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -9.64760000
NELEM =	197	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000
NELEM =	198	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000
NELEM =	199	NUCLO =	2
LPOIN =	8	LDOFN =	3 LOAD = -3.61780000
LPOIN =	7	LDOFN =	3 LOAD = -3.61780000



NELEM =	200	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -3.61780000
LPOIN =	7	LDOFN =	3	LOAD = -1.80890000
NELEM =	201	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
LPOIN =	7	LDOFN =	3	LOAD = -12.05960000
NELEM =	202	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
LPOIN =	7	LDOFN =	3	LOAD = -12.05960000
NELEM =	203	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
LPOIN =	7	LDOFN =	3	LOAD = -12.05960000
NELEM =	204	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
LPOIN =	7	LDOFN =	3	LOAD = -12.05960000
NELEM =	205	NUCLO =	2	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
LPOIN =	7	LDOFN =	3	LOAD = -12.05960000
NELEM =	206	NUCLO =	1	
LPOIN =	8	LDOFN =	3	LOAD = -12.05960000
NELEM =	207	NUCLO =	3	
LPOIN =	1	LDOFN =	3	LOAD = -1.80890000

LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	208	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	209	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	210	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-9.64760000
NELEM =	211	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	212	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	213	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-3.61780000
NELEM =	214	NUCLO =	2		
LPOIN =	8	LDOFN =	3	LOAD =	-3.61780000
LPOIN =	7	LDOFN =	3	LOAD =	-1.80890000



```

*** INCREMENT NUMBER      2
O  LOAD FACTOR = .20000
TOTAL BEBAN EKSTERNAL = -.5209695000E+05
O  DISPLACEMENTS
O  NODE      X-DISP      Y-DISP      Z-DISP      AF-ROT      BT-ROT
1      .000000E+00      .601457E+00      -.652246E+00      .863140E-06      .403917E-07
2      .848277E+00      .601461E+00      -.652791E+00      .863152E-06      .404258E-07
3      .848274E+00      .601464E+00      -.653336E+00      .863163E-06      .404599E-07
4      .848271E+00      .601468E+00      -.653340E+00      .863174E-06      .404940E-07
5      .848267E+00      .601471E+00      -.653884E+00      .863186E-06      .405281E-07
6      .848264E+00      .601475E+00      -.653887E+00      .863197E-06      .405622E-07
7      .848260E+00      .601478E+00      -.654431E+00      .863209E-06      .405963E-07
8      .848257E+00      .601481E+00      -.654974E+00      .863220E-06      .406304E-07
9      .848254E+00      -.601485E+00      -.654978E+00      .863231E-06      -.402554E-07
10     .848250E+00      -.601488E+00      -.655520E+00      .863243E-06      -.402440E-07
11     .848247E+00      -.601492E+00      -.656062E+00      .863254E-06      -.402326E-07
12     .848243E+00      -.601495E+00      -.656066E+00      .863265E-06      -.402213E-07
13     .848240E+00      -.601498E+00      -.656608E+00      .863277E-06      -.402099E-07
14     .848237E+00      -.601502E+00      -.657149E+00      .863288E-06      -.401985E-07
15     .848233E+00      -.601505E+00      -.657152E+00      .863299E-06      -.401872E-07
16     .848230E+00      -.601509E+00      -.657693E+00      .863311E-06      -.401758E-07
17     .000000E+00      -.601512E+00      -.658233E+00      .863322E-06      -.401645E-07
18     .000000E+00      .601515E+00      .000000E+00      .854872E-06      .409713E-07
19     .845681E+00      .601519E+00      -.628322E+00      .854883E-06      .410054E-07
20     .845678E+00      .601522E+00      -.628365E+00      .854895E-06      .410395E-07
21     .845674E+00      .601526E+00      -.628368E+00      .854906E-06      .410736E-07
22     .845671E+00      -.601529E+00      -.628242E+00      .854917E-06      -.401076E-07
23     .845667E+00      -.601532E+00      -.628277E+00      .854929E-06      -.400963E-07
24     .845664E+00      -.601536E+00      -.628273E+00      .854940E-06      -.400849E-07
25     .845661E+00      -.601539E+00      -.628308E+00      .854952E-06      -.400735E-07
26     .000000E+00      -.601543E+00      .000000E+00      .854963E-06      -.400622E-07
27     .000000E+00      .597516E+00      .000000E+00      .846677E-06      .278452E-07
28     .843161E+00      .597520E+00      -.617310E+00      .846688E-06      .278793E-07
29     .843158E+00      .597523E+00      -.617332E+00      .846700E-06      .279134E-07
30     .843154E+00      .597526E+00      -.617355E+00      .846711E-06      .279475E-07
31     .843151E+00      .597530E+00      -.617359E+00      .846723E-06      .279816E-07
32     .843148E+00      .597533E+00      -.617381E+00      .846734E-06      .280157E-07
33     .843144E+00      .597537E+00      -.617404E+00      .846745E-06      .280497E-07

```

34	.843141E+00	.597540E+00	-.617408E+00	.846757E-06	.280838E-07
35	.843137E+00	-.597544E+00	-.617162E+00	.846768E-06	-.265270E-07
36	.843134E+00	-.597547E+00	-.617177E+00	.846779E-06	-.265156E-07
37	.843131E+00	-.597550E+00	-.617173E+00	.846791E-06	-.265043E-07
38	.843127E+00	-.597554E+00	-.617188E+00	.846802E-06	-.264929E-07
39	.843124E+00	-.597557E+00	-.617203E+00	.846813E-06	-.264816E-07
40	.843120E+00	-.597561E+00	-.617199E+00	.846825E-06	-.264702E-07
41	.843117E+00	-.597564E+00	-.617214E+00	.846836E-06	-.264588E-07
42	.843113E+00	-.597567E+00	-.617229E+00	.846848E-06	-.264475E-07
43	.000000E+00	-.597571E+00	.000000E+00	.846859E-06	-.264361E-07
44	.000000E+00	.595559E+00	.000000E+00	.838733E-06	.217083E-07
45	.840662E+00	.595563E+00	-.614143E+00	.838744E-06	.217424E-07
46	.840659E+00	.595566E+00	-.614146E+00	.838755E-06	.217765E-07
47	.840655E+00	.595570E+00	-.614162E+00	.838767E-06	.218106E-07
48	.840652E+00	-.595573E+00	-.613810E+00	.838778E-06	-.196628E-07
49	.840648E+00	-.595576E+00	-.613805E+00	.838790E-06	-.196515E-07
50	.840645E+00	-.595580E+00	-.613814E+00	.838801E-06	-.196401E-07
51	.840642E+00	-.595583E+00	-.613822E+00	.838812E-06	-.196287E-07
52	.000000E+00	-.595587E+00	.000000E+00	.838824E-06	-.196174E-07
53	.000000E+00	.593575E+00	.000000E+00	.830852E-06	.152987E-07
54	.838237E+00	.593578E+00	-.613447E+00	.830864E-06	.153328E-07
55	.838233E+00	.593582E+00	-.613460E+00	.830875E-06	.153668E-07
56	.838230E+00	.593585E+00	-.613463E+00	.830886E-06	.154009E-07
57	.838226E+00	.593589E+00	-.613476E+00	.830898E-06	.154350E-07
58	.838223E+00	.593592E+00	-.613488E+00	.830909E-06	.154691E-07
59	.838220E+00	.593596E+00	-.613492E+00	.830921E-06	.155032E-07
60	.838216E+00	.593599E+00	-.613504E+00	.830932E-06	.155373E-07
...					
650	-.787517E+00	.616642E+00	-.824997E+00	.645798E-06	.105756E-06
651	-.787520E+00	.616645E+00	-.825000E+00	.645786E-06	.105790E-06
652	-.783078E+00	-.616648E+00	-.825004E+00	.630957E-06	-.761873E-07
653	-.783075E+00	-.616652E+00	-.825007E+00	.630923E-06	-.761760E-07
654	-.783072E+00	-.616655E+00	-.825011E+00	.630889E-06	-.761646E-07
655	-.783068E+00	-.616659E+00	-.825014E+00	.630854E-06	-.761532E-07
656	-.783065E+00	-.616662E+00	-.825017E+00	.630820E-06	-.761419E-07
657	-.783061E+00	-.616665E+00	-.825021E+00	.630786E-06	-.761305E-07
658	-.783058E+00	-.616669E+00	-.825024E+00	.630752E-06	-.761192E-07



659	-.783055E+00	-.616672E+00	-.825028E+00	.630718E-06	-.761078E-07
660	-.783051E+00	-.616676E+00	-.825031E+00	.630684E-06	-.760964E-07
661	-.783048E+00	-.616679E+00	-.825034E+00	.630650E-06	-.760851E-07
662	-.783044E+00	-.616682E+00	-.825038E+00	.630616E-06	-.760737E-07
663	-.783041E+00	-.616686E+00	-.825041E+00	.630582E-06	-.760623E-07
664	-.783037E+00	-.616689E+00	-.825045E+00	.630548E-06	-.760510E-07
665	-.783034E+00	-.616693E+00	-.825048E+00	.630514E-06	-.760396E-07
666	-.783031E+00	-.616696E+00	-.825051E+00	.630479E-06	-.760282E-07
667	-.783027E+00	-.616700E+00	-.825055E+00	.630445E-06	-.760169E-07
668	-.783024E+00	-.616703E+00	-.825058E+00	.630411E-06	-.760055E-07
669	-.783020E+00	-.616706E+00	-.825062E+00	.630377E-06	-.759942E-07
670	-.783017E+00	-.616710E+00	-.825065E+00	.630343E-06	-.759828E-07
671	-.783014E+00	-.616713E+00	-.825069E+00	.630309E-06	-.759714E-07
672	-.783010E+00	-.616717E+00	-.825072E+00	.630275E-06	-.759601E-07
673	-.783007E+00	-.616720E+00	-.825075E+00	.630241E-06	-.759487E-07
674	-.783003E+00	-.616723E+00	-.825079E+00	.630207E-06	-.759373E-07
675	-.783000E+00	-.616727E+00	-.825082E+00	.630173E-06	-.759260E-07
676	-.782997E+00	-.616730E+00	-.825086E+00	.630139E-06	-.759146E-07
677	-.782993E+00	-.616734E+00	-.825089E+00	.630104E-06	-.759032E-07
678	-.782990E+00	-.616737E+00	-.825092E+00	.630070E-06	-.758919E-07
679	-.782986E+00	-.616740E+00	-.825096E+00	.630036E-06	-.758805E-07
680	-.782983E+00	-.616744E+00	-.825099E+00	.630002E-06	-.758692E-07
681	-.782979E+00	-.616747E+00	-.825103E+00	.629968E-06	-.758578E-07
682	-.782976E+00	-.616751E+00	-.825106E+00	.629934E-06	-.758464E-07
683	-.782973E+00	-.616754E+00	-.825109E+00	.629900E-06	-.758351E-07
684	-.782969E+00	-.616757E+00	-.825113E+00	.629866E-06	-.758237E-07
685	-.782966E+00	-.616761E+00	-.825116E+00	.629832E-06	-.758123E-07
686	-.782962E+00	-.616764E+00	-.825120E+00	.629798E-06	-.758010E-07
687	-.782959E+00	-.616768E+00	-.825123E+00	.629764E-06	-.757896E-07
688	-.782956E+00	-.616771E+00	-.825126E+00	.629729E-06	-.757782E-07
689	-.782952E+00	-.616775E+00	-.825130E+00	.629695E-06	-.757669E-07
690	-.782949E+00	-.616778E+00	-.825133E+00	.629661E-06	-.757555E-07
691	-.782945E+00	-.616781E+00	-.825137E+00	.629627E-06	-.757441E-07
692	-.782942E+00	-.616785E+00	-.825140E+00	.629593E-06	-.757328E-07





# **TUGAS AKHIR**

## **LAMPIRAN 5**

**CONTOH OUTPUT DATA  
TEGANGAN DAN RESULTAN GAYA**



# LAMPIRAN 5. CONTOH OUTPUT DATA TEGANGAN DAN RESULTAN GAYA

```

O      STRESSES
O KLAYR      KK-STR      YY-STR      XY-STR      XZ-STR      YZ-STR      EFF.STRESS      EFF.PL.STRAIN

O *      ELEMENT NO. = 161

      G.P. NO. = 1
      1      .287942E+03      .943136E+02      -.107971E+02      .760892E+01      -.501842E+02      .301095E+03      .200814E-03

      STRESS RESULTANTS =      N-XX = .14973E-01      M-XX = .38929E-01
                               N-YY = .49043E-02      M-YY = .12751E-01
                               N-XY = -.56145E-03      M-XY = -.14598E-02
                               Q-XZ = .39566E-03      Q-YZ = -.26096E-02

      G.P. NO. = 2
      1      -.852004E+01      -.216881E+01      .511106E+02      .946252E+02      .141481E+03      .301094E+03      .194975E-03

      STRESS RESULTANTS =      N-XX = -.44304E-03      M-XX = -.11519E-02
                               N-YY = -.11278E-03      M-YY = -.29322E-03
                               N-XY = .26577E-02      M-XY = .69101E-02
                               Q-XZ = .49205E-02      Q-YZ = .73570E-02

      G.P. NO. = 3
      1      .241681E+03      .792173E+02      -.347266E+02      -.904883E+02      -.643040E+02      .301095E+03      .198948E-03

      STRESS RESULTANTS =      N-XX = .12567E-01      M-XX = .32675E-01
                               N-YY = .41193E-02      M-YY = .10710E-01
                               N-XY = -.18058E-02      M-XY = -.46950E-02
                               Q-XZ = -.47054E-02      Q-YZ = -.33438E-02

      G.P. NO. = 4
      1      .262934E+03      .812301E+02      .783256E+02      -.627872E+02      -.846623E+01      .301094E+03      .193345E-03

      STRESS RESULTANTS =      N-XX = .13673E-01      M-XX = .35548E-01
                               N-YY = .42240E-02      M-YY = .10982E-01
                               N-XY = .40729E-02      M-XY = .10589E-01
                               Q-XZ = -.32649E-02      Q-YZ = -.44024E-03

O *      ELEMENT NO. = 162
    
```

G.P. NO. = 1  
1    -.621483E+02    -.427348E+02    -.386356E+02    .104820E+03    -.133438E+03    .301095E+03    .197086E-03

STRESS RESULTANTS =    N-XX = -.32317E-02    M-XX = -.84024E-02  
                              N-YY = -.22222E-02    M-YY = -.57777E-02  
                              N-XY = -.20090E-02    M-XY = -.52235E-02  
                              Q-XZ = .54506E-02    Q-YZ = -.69388E-02

G.P. NO. = 2  
1    .163200E+03    .506962E+02    .997614E+02    .116627E+02    .115123E+03    .301094E+03    .187052E-03

STRESS RESULTANTS =    N-XX = .84864E-02    M-XX = .22064E-01  
                              N-YY = .26362E-02    M-YY = .68540E-02  
                              N-XY = .51876E-02    M-XY = .13488E-01  
                              Q-XZ = .60646E-03    Q-YZ = .59864E-02

G.P. NO. = 3  
1    .256811E+03    .608874E+02    -.555104E+02    -.699969E+02    .527053E+02    .301094E+03    .195129E-03

STRESS RESULTANTS =    N-XX = .13354E-01    M-XX = .34720E-01  
                              N-YY = .31661E-02    M-YY = .82319E-02  
                              N-XY = -.28865E-02    M-XY = -.75049E-02  
                              Q-XZ = -.36398E-02    Q-YZ = .27407E-02

G.P. NO. = 4  
1    .143806E+03    .437588E+02    .957746E+02    -.503756E+02    .116067E+03    .301094E+03    .187605E-03

STRESS RESULTANTS =    N-XX = .74779E-02    M-XX = .19442E-01  
                              N-YY = .22755E-02    M-YY = .59161E-02  
                              N-XY = .49803E-02    M-XY = .12949E-01  
                              Q-XZ = -.26195E-02    Q-YZ = .60355E-02

O \* ELEMENT NO. = 163

G.P. NO. = 1  
1    .123673E+03    .519570E+02    .916095E+02    .128059E+02    .135372E+03    .301094E+03    .187824E-03

STRESS RESULTANTS =    N-XX = .64310E-02    M-XX = .16720E-01  
                              N-YY = .27018E-02    M-YY = .70245E-02  
                              N-XY = .47637E-02    M-XY = .12385E-01  
                              Q-XZ = .66591E-03    Q-YZ = .70393E-02



G.P. NO. = 2  
1 .236512E+02 .181613E+02 .105561E+03 -.706680E+01 .142163E+03 .301093E+03 .184100E-03

STRESS RESULTANTS = N-XX = .12299E-02 M-XX = .31976E-02  
N-YY = .94439E-03 M-YY = .24554E-02  
N-XY = .54892E-02 M-XY = .14272E-01  
Q-XZ = -.36747E-03 Q-YZ = .73924E-02

G.P. NO. = 3  
1 .746380E+02 .318496E+02 .738336E+02 .775368E+01 .156117E+03 .301094E+03 .189732E-03

STRESS RESULTANTS = N-XX = .38812E-02 M-XX = .10091E-01  
N-YY = .16562E-02 M-YY = .43060E-02  
N-XY = .38393E-02 M-XY = .99822E-02  
Q-XZ = .40319E-03 Q-YZ = .81181E-02

G.P. NO. = 4  
1 -.328044E+02 .249683E+02 .103552E+03 -.185841E+02 .140343E+03 .301093E+03 .182940E-03

STRESS RESULTANTS = N-XX = -.17058E-02 M-XX = -.44351E-02  
N-YY = .12983E-02 M-YY = .33757E-02  
N-XY = .53847E-02 M-XY = .14000E-01  
Q-XZ = -.96637E-03 Q-YZ = .72978E-02

O \* ELEMENT NO. = 164

G.P. NO. = 1  
1 .149514E+03 .652320E+02 -.461678E+02 -.372158E+02 -.104713E+03 .301093E+03 .181903E-03

STRESS RESULTANTS = N-XX = .77747E-02 M-XX = .20214E-01  
N-YY = .33921E-02 M-YY = .88193E-02  
N-XY = -.24007E-02 M-XY = -.62418E-02  
Q-XZ = -.19352E-02 Q-YZ = -.54450E-02

G.P. NO. = 2  
1 -.549546E+01 .942937E+01 .130723E+03 .144440E+02 .119376E+03 .301092E+03 .174891E-03

STRESS RESULTANTS = N-XX = -.28576E-03 M-XX = -.74298E-03  
N-YY = .49033E-03 M-YY = .12748E-02  
N-XY = .67976E-02 M-XY = .17674E-01  
Q-XZ = .75109E-03 Q-YZ = .62076E-02

G.P. NO. = 3

1    -.644874E+02    .170298E+02    .151326E+03    -.236994E+02    .789371E+02    .301091E+03    .165927E-03

STRESS RESULTANTS =    N-XX = -.33533E-02    M-XX = -.87186E-02  
                               N-YY = .88555E-03    M-YY = .23024E-02  
                               N-XY = .78690E-02    M-XY = .20459E-01  
                               Q-XZ = -.12324E-02    Q-YZ = .41047E-02

G.P. NO. =    4

1    -.872143E+01    -.515456E+02    -.131981E+03    .294563E+02    -.111596E+03    .301092E+03    .169184E-03

STRESS RESULTANTS =    N-XX = -.45351E-03    M-XX = -.11791E-02  
                               N-YY = -.26804E-02    M-YY = -.69689E-02  
                               N-XY = -.68630E-02    M-XY = -.17844E-01  
                               Q-XZ = .15317E-02    Q-YZ = -.58030E-02

O \*    ELEMENT NO. = 165

G.P. NO. =    1

1    .629359E+02    -.138595E+03    -.133992E+03    .527441E+02    .868325E+01    .301091E+03    .164031E-03

STRESS RESULTANTS =    N-XX = .32727E-02    M-XX = .85088E-02  
                               N-YY = -.72069E-02    M-YY = -.18738E-01  
                               N-XY = -.69676E-02    M-XY = -.18115E-01  
                               Q-XZ = .27427E-02    Q-YZ = .45153E-03

G.P. NO. =    2

1    .649851E+02    .741841E+02    .163460E+03    .274600E+02    .441787E+02    .301092E+03    .166627E-03

STRESS RESULTANTS =    N-XX = .33792E-02    M-XX = .87859E-02  
                               N-YY = .38576E-02    M-YY = .10030E-01  
                               N-XY = .84999E-02    M-XY = .22100E-01  
                               Q-XZ = .14279E-02    Q-YZ = .22973E-02

G.P. NO. =    3

1    -.971900E+02    -.842640E+02    -.118253E+03    .629100E+02    .997594E+02    .301093E+03    .179829E-03

STRESS RESULTANTS =    N-XX = -.50539E-02    M-XX = -.13140E-01  
                               N-YY = -.43817E-02    M-YY = -.11392E-01  
                               N-XY = -.61492E-02    M-XY = -.15988E-01  
                               Q-XZ = .32713E-02    Q-YZ = .51875E-02

G.P. NO. =    4

1    .307188E+02    .229612E+03    .904819E+02    -.276580E+01    -.772386E+02    .301093E+03    .178968E-03



STRESS RESULTANTS = N-XX = .15974E-02 M-XX = .41531E-02  
 N-YY = .11940E-01 M-YY = .31043E-01  
 N-XY = .47051E-02 M-XY = .12233E-01  
 Q-XZ = -.14382E-03 Q-YZ = -.40164E-02

O \* ELEMENT NO. = 166

G.P. NO. = 1

1 .299440E+01 -.128420E+03 -.457631E+02 .876883E+02 .118096E+03 .301093E+03 .184757E-03

STRESS RESULTANTS = N-XX = .15571E-03 M-XX = .40484E-03  
 N-YY = -.66778E-02 M-YY = -.17362E-01  
 N-XY = -.23797E-02 M-XY = -.61871E-02  
 Q-XZ = .45598E-02 Q-YZ = .61410E-02

G.P. NO. = 2

1 .211566E+02 .843008E+01 .168127E+03 .289595E+02 -.485951E+02 .301091E+03 .162846E-03

STRESS RESULTANTS = N-XX = .11001E-02 M-XX = .28603E-02  
 N-YY = .43836E-03 M-YY = .11397E-02  
 N-XY = .87426E-02 M-XY = .22730E-01  
 Q-XZ = .15059E-02 Q-YZ = -.25269E-02

G.P. NO. = 3

1 -.136169E+03 -.158587E+03 -.764531E+02 .316663E+02 .122261E+03 .301094E+03 .193808E-03

STRESS RESULTANTS = N-XX = -.70808E-02 M-XX = -.18410E-01  
 N-YY = -.82465E-02 M-YY = -.21441E-01  
 N-XY = -.39756E-02 M-XY = -.10336E-01  
 Q-XZ = .16466E-02 Q-YZ = .63576E-02

G.P. NO. = 4

1 -.185438E+03 .337514E+02 .548281E+02 .173979E+02 -.117136E+03 .301093E+03 .183469E-03

STRESS RESULTANTS = N-XX = -.96428E-02 M-XX = -.25071E-01  
 N-YY = .17551E-02 M-YY = .45631E-02  
 N-XY = .28511E-02 M-XY = .74127E-02  
 Q-XZ = .90469E-03 Q-YZ = -.60910E-02

O \* ELEMENT NO. = 167

G.P. NO. = 1

1    -.212795E+03    .847634E+02    -.226991E+02    .817545E+02    .210700E+02    .301093E+03    .181668E-03

STRESS RESULTANTS =    N-XX = -.11065E-01    M-XX = -.28769E-01  
                               N-YY = .44077E-02    M-YY = .11460E-01  
                               N-XY = -.11804E-02    M-XY = -.30689E-02  
                               Q-XZ = .42512E-02    Q-YZ = .10956E-02

G.P. NO. =    2

1    -.289295E+03    -.966368E+02    -.290204E+02    .698377E+02    -.223411E+01    .301095E+03    .200150E-03

STRESS RESULTANTS =    N-XX = -.15043E-01    M-XX = -.39112E-01  
                               N-YY = -.50251E-02    M-YY = -.13065E-01  
                               N-XY = -.15091E-02    M-XY = -.39235E-02  
                               Q-XZ = .36316E-02    Q-YZ = -.11617E-03

G.P. NO. =    3

1    -.446598E+02    .230699E+02    -.168812E+03    .417250E+02    -.129616E+02    .301092E+03    .166816E-03

STRESS RESULTANTS =    N-XX = -.23223E-02    M-XX = -.60379E-02  
                               N-YY = .11996E-02    M-YY = .31190E-02  
                               N-XY = -.87782E-02    M-XY = -.22823E-01  
                               Q-XZ = .21697E-02    Q-YZ = -.67400E-03

G.P. NO. =    4

1    -.240035E+03    -.873239E+02    .109416E+03    .430441E+02    .241745E+01    .301094E+03    .186327E-03

STRESS RESULTANTS =    N-XX = -.12482E-01    M-XX = -.32452E-01  
                               N-YY = -.45408E-02    M-YY = -.11806E-01  
                               N-XY = .56896E-02    M-XY = .14793E-01  
                               Q-XZ = .22383E-02    Q-YZ = .12571E-03

O \*    ELEMENT NO. = 168

G.P. NO. =    1

1    -.303176E+03    -.860985E+02    .957148E+01    .511583E+02    .150152E+02    .301095E+03    .199451E-03

STRESS RESULTANTS =    N-XX = -.15765E-01    M-XX = -.40989E-01  
                               N-YY = -.44771E-02    M-YY = -.11640E-01  
                               N-XY = .49772E-03    M-XY = .12940E-02  
                               Q-XZ = .26602E-02    Q-YZ = .78079E-03

G.P. NO. =    2

1    -.282926E+03    -.172304E+03    -.613624E+02    -.600640E+01    .405641E+02    .301095E+03    .200847E-03



STRESS RESULTANTS = N-XX = -.14712E-01 M-XX = -.38251E-01  
 N-YY = -.89598E-02 M-YY = -.23295E-01  
 N-XY = -.31908E-02 M-XY = -.82961E-02  
 Q-XZ = -.31233E-03 Q-YZ = .21093E-02

G.P. NO. = 3

1 -.228856E+03 -.572000E+02 -.111879E+03 .499692E+02 -.102202E+02 .301093E+03 .183322E-03

STRESS RESULTANTS = N-XX = -.11900E-01 M-XX = -.30941E-01  
 N-YY = -.29744E-02 M-YY = -.77333E-02  
 N-XY = -.58177E-02 M-XY = -.15126E-01  
 Q-XZ = .25984E-02 Q-YZ = -.53145E-03

G.P. NO. = 4

1 -.114812E+03 -.735382E+02 .157561E+03 .412793E+02 .263794E+02 .301092E+03 .171512E-03

STRESS RESULTANTS = N-XX = -.59702E-02 M-XX = -.15522E-01  
 N-YY = -.38240E-02 M-YY = -.99422E-02  
 N-XY = .81931E-02 M-XY = .21302E-01  
 Q-XZ = .21465E-02 Q-YZ = .13717E-02